[FALL 2025 CS485/585]
Project report: Differential Cryptanalysis against 1-Round DES

Jarren Calizo Benjamin Chong Wesley Grzemkowski

Department of Computer Science
Portland State University

{*calizo,chongben,wesleyg*}*@pdx.edu*

December 10, 2025

Abstract

An executive summary of your report.

Your report should be about 10 pages without counting this title page or the references in the end. Give a comprehensive account but concisely.

1 Introduction

DES is a 16-round encryption scheme that operates on a 64-bit message using a 56-bit key. The full 16-round DES scheme has proven rather secure considering its limited key space. It has however fallen out of use since the relatively small key space makes the scheme vulnerable to brute-force attacks. From what we have found in our research, reducing the number of rounds DES uses makes it vulnerable to differential cryptanalysis attacks.

In this report, we will show the attack we implemented to recover the key for a one-round DES scheme. We will also show how such an attack can be used to recover the key for DES with a higher round count with relative efficiency.

Organization

The remainder of the paper is organized as follows: ...

2 Preliminaries

This section provides preliminary materials, such as summarizing common notations, definitions, and useful facts for the main body.

2.1 Notation

We use the following notations in the discussion of our attack.

Numbers: To differentiate between hexadecimal, decimal, and binary number a subscript is used to denote the number's base. Decimal numbers are denoted with no subscript. (e.g., $31 = 1F_{16} = 00011111_2$)

Plaintext and Ciphertext: The plaintext is denoted by M, and the ciphertext is denoted by C. Because the initial permutation has little bearing on our attack model, M represents the plaintext after the initial permutation, and C represents the ciphertext before the final permutation.

Feistel Rounds: The state of the ciphertext after round is i denoted by the pair (L_i, R_i), where L_i denotes the leftmost 32-bits and R_i denotes the rightmost 32-bits. The pair (L_0, R_0) is used to denote the input to the first round of the feistel network.

Differences: In differential cryptanalysis, variables are usually considered in pairs. Given a value X, its corresponding value in the pair is denoted by X'. The difference of X and X' is defined as $X \oplus X'$, and this value is denoted by ΔX.

Subkeys: The subkeys is denoted by K_i, where i indicates the round in which the subkey is used. The notation $K^{(j)}$ is used to denote the j-th 6-bit chunk of K.

Sets: The set of all possible plaintexts is denoted by $\mathcal{M}$. The set of all possible ciphertexts is denoted by $\mathcal{C}$. The set of all possible subkeys is denoted by $\mathcal{K}$.

Characteristic: The notation (M^*, C^*) is used to denote a differential characteristic, i.e., a difference $(\Delta M, \Delta C)$ which occurs with some known probability $\mathbb{P}(\Delta C \mid \Delta M)$.

Initial Permutation: The initial permutation of DES is denoted by $IP(X)$, and the inverse initial permutation, or final permutation, is denoted by $IP^{-1}(X)$.

F Function: The F function, or mangler function, is denoted by $F(X)$.

Expansion Function: The expansion function is denoted by $E(X)$.

P Permutation: The P permutation is denoted by $P(X)$.

S-boxes: The collective 48-bit to 32-bit S-box layer is denoted by the function $S(X)$. The individual eight S-boxes are denoted by $S^1, S^2, \ldots, S^8$.

Intermediary Values: The output of the expansion function is denoted by I. The input to the S-box layer is denoted by B. The output of the S-box layer is denoted by Y. A superscript (j) is used to denote the j-th 6-bit chunk, or 4-bit chunk.

3 DES Overview

3.1 Key Generation

Each of the 16 rounds of DES uses its own 48-bit subkey. Each of these subkeys are obtained by inputting a single 64-bit key into the key generation algorithm. Note that the true keyspace of DES is 56 bits. When a 64-bit key is given as input, the 8th bit of each byte is not used. The 56 bits that are used are then passed through a permutation. This permutation is the same for all keys. The left half of the output of this permutation becomes C_0 and the right half becomes D_0. We then use a series of left shifts on C_0 and D_0 to obtain C_n and D_n for $1 <= n <= 16$. for $n = 1, 2, 9, 16$ C_n and D_n are a single left shift from C_{n-1} and D_{n-1}, and for the other values of n, two left shifts are used. We then have sixteen 56-bit strings $C_n D_n$ for $1 <= n <= 16$. To obtain our 48-bit subkeys K_n we apply a permutation to 48 bits of the corresponding $C_n D_n$. For reduced round DES of n rounds, we can just use the first n keys.

3.2 Encryption Overview

To encrypt a 64 bit message under DES, the message is first passed through a set initial permutation IP that rearranges the bits. This permutation is the same for every encryption. The output of this permutation becomes the input to the first round. The output of that round becomes the input for the second round. This continues until the output of the sixteenth round, which is run through the inverse of IP. The output of that inverse permutation is the ciphertext.

For each of the rounds, the input is split into a right half R and a left half L each a 32-bit string. Each round uses the function F_K where K is the subkey for that round. The left half of the rounds output is $F_K(R) \oplus L$ and the right half is simply L.

images/des_rounds.png

Figure 3.1: 4-Round DES

3.3 F Function

The F function, also called the mangler function, is designed to produce a pseudorandom 32-bit output

from a 32-bit input and 48-bit key. The input to
F is runthrough the expansion box E, producing a 48-bit string, which is then XOR-ed with the key. The output of the XOR is split into 8 6-bit strings. These strings become the input for the 8 S-boxes. Each S-box is a unique mapping of 6-bit strings to 4-bit strings. After the S-box layer, we combine the eight resulting 4-bit strings into a 32-bit string and use the P permutation to produce the 32-bit output of the F function.

images/des_f_function.png

3.4 Vulnerabilities

Since The expansion box and P permutation in the F function are predetermined, much of the strength of this scheme relies on the S boxes. For our attack we focus on the XORs of pairs of messages, and the corresponding XORs of the pairs of ciphertexts. The problem with the S boxes is that for a given input XOR, some output XORs are more likely to coour than others. We can use this to narrow down the possible keys by choosing certain plaintext pairs with certain XORs.

4 Attack Against 1-Round DES

Pick appropriate titles and subtitles according to the topic of your project.

4.1 The Attack Model

4.1.1 1-Round DES

The attack shown is a chosen-plaintext against 1-round DES. The reduction to 1-round DES was done to demonstrate the impact of differential cryptanalysis on DES. This attack model is not intended to be a practical consideration. Our goal is to instead use 1-round DES to simulate a single round of more complex encryption algorithm, and we have adjusted our attack model accordingly.

To understand why, first note that 1-round DES is trivial to construct an attack against without any differential cryptanalysis. Consider a known plaintext-ciphertext pair $M = (L_0, R_0)$ and $C = (L_1, R_1)$. With this pair, an adversary computes $I = E(R_0)$ and $Y = P^{-1}(R_1 \oplus L_0)$. The adversary then computes $I^{(j)} \oplus B^{(j)} = K^{(j)}$ for all possiblilities of $B^{(j)}$ and is left with only 4 potential values for $K^{(j)}$. After repeating this for $j = 1, \ldots, 8$ the size of the key space is reduced to only $4^8 = 2^{16}$.

Now, consider an n-round DES scheme with a plaintext-cipertext pair $M = (L_0, R_0)$ and $C = (L_n, R_n)$. Note that $R_n = L_{n-1} \oplus F(L_n, K)$ and only L_n and R_n are known to the adversary. If the adversary knows L_{n-1}, they can learn $F(L_n, K)$ and use the attack previously described. This was only possible in 1-round DES because $L_0 = L_{n-1}$. Because of this, in our attack model we prohibit the adversary from using the actual value of M after sending them to encryption oracle. How this exactly works will become more clear as we descibe our attack.

In our model, we consider the attack to be successful if it can output the first subkey K_1, with a high probability of correctness. In a complete DES scheme, this would be similar to recovering only the final subkey. We are justified in this approach because it is simple to derive bits in the actual key even from a single subkey.

4.1.2 General Attack

The goal of our attack is to generate a number ciphertext pairs (C, C') where ΔB and ΔY are known to the adversary. Using such a ciphertext pair, an adversary can use L_1 to compute a subset of possible subkeys able to produce this result. Given a enough of these ciphertext pairs, an adversary can reduce the subkey space considerably, and it becomes possible to brute-force the exact subkey. The general algorithm is as follows:

1. Select a characteristic (B^*, Y^*) which occurs with high probability in S.

2. Generate a set of plaintext pairs $\mathbf{M}$ where $\Delta I = B^*$.

3. Encrypt the all pairs in $\mathbf{M}$ to get a set of ciphertext pairs $\mathbf{C}$.

4. Derive a new set $\mathbf{C}^*$ which only contains pairs in $\mathbf{C}$ where $\Delta Y = Y^*$.

5. For each $(C, C') \in \mathbf{C}^*$, compute all possible subkeys and count the number of times each subkey appears.

6. Let $\mathbf{K}$ be the set of all subkeys in $\mathcal{K}$ which occurs with the highest frequency.

After this algorithm has been completed, it will always be true that $K \in \mathbf{K}$. Supposing that $\mathbf{C}^*$ is of sufficient size and the characteristic (B^*, Y^*) is chosen adequetly, $\mathbf{K}$ can be hypothetically as small as 2^8 making it a trivial operation to brute force.

4.2 Description of Attack

4.2.1 Choosing Characteristics

For our attack, it is desireable for our characteristic (B^*, Y^*) to occur with the highest possible probability, i.e., for $\mathbb{P}(S(B) \oplus S(B \oplus B^*) = Y^*)$ given a uniformly random B. This is because it will increase the likelihood for any given pair $(C, C') \in \mathbf{C}$ that $\Delta Y = Y^*$, and produce a $\mathbf{C}^*$ of sufficient size with the least number of encryptions. Simultaneously, it is also desireable for $\mathbf{C}^*$ to diminish the size of $\mathbf{K}$ by as much as possible.

Generally speaking, if we choose B^* such that it only affects a single S-box, i.e., for some S-box s, $B^{*(j)} = 00\,00\,00_2$ for all $j \in \{1, \ldots, 8 \mid j \neq s\}$, we maximize the value of $\mathbb{P}(Y^* \mid B^*)$. However when $\Delta B^{(j)} = 0$, it means that any input B to S^s will produce the same difference $\Delta O^{(j)} = 0$. So, all subkeys $K^{(j)}$ are equally probable given a known I and $\mathcal{K}^{(j)} = \mathbf{K}^{(j)}$.

For our attack, we chose to prioritize maximizing the probability of our characteristic occuring by only targeting a single S-box at a time. This means that 8 characteristics must be constructed to reduce the size of $\mathbf{K}^{*(j)}$ for each $j \in 1, \ldots, 8$. To do this, we are forced to find characteristics $B^{*(j)}$ such that they follow the pattern $00\,xx\,00_2$ as only the middle two bits of each 6-bit chunk are not duplicated by the expansion function. Since we cannot choose $B^{*(j)} = 00\,00\,00_2$ this leaves us with only 3 potential input characteristics for each S-box, making only $3 \cdot 2^4 = 48$ usable characteristics of the over 1000 total characteristics present in each S-box.

We selected the highest probability characteristic from these 48 to get the characteristics shown in Table **??** for each S-box. As shown, this still gives us characteristics with relatively high probabilities. If we expand our choices to include all characteristics, the best we can do is a count of 16/64 or a 0.25 probability of occurence. Even with our constraints, we still are left with $\mathbb{P}(Y^* \mid B^*) \approx 20.7\%$ which is easily sufficient for our purposes. It is highly likely that better characteristics exist, and provide higher fidelity while retaining a similar differential probability.

4.3 Generating Plaintext Ciphertext Pairs

Once we have chosen a characteristic, the next step is to generating our plaintext and ciphertext pairs. If we already have a desired M^*, we can easily generate our set $\mathbf{M}$

S-box	Input	Output	Count	Probability
S^1	$0c_{16}$	e_{16}	14/64	21.88%
S^2	08_{16}	a_{16}	16/64	25.00%
S^3	04_{16}	9_{16}	12/64	18.75%
S^4	04_{16}	6_{16}	12/64	18.75%
S^5	04_{16}	6_{16}	10/64	15.62%
S^6	08_{16}	6_{16}	16/64	25.00%
S^7	$0c_{16}$	c_{16}	14/64	21.88%
S^8	04_{16}	7_{16}	12/64	18.75%

Table 4.1: Chosen Characteristics

4.4 Reducing Key Space

5 Translating to Multi-Round DES

6 Something more

7 Conclusion

This project implemented and experimentally evaluated a differential cryptanalysis attack on a one-round instance of DES, with the goal of recovering the last-round subkey from chosen plaintext-ciphertext pairs. By constructing S-box difference distribution tables, generating structured plaintext pairs with fixed input differences, and aggregating votes over candidate subkeys, the attack consistently reduced the 48-bit last-round key space from 2^{48} possibilities down to roughly 2^{10}-2^{12} candidates, after which a small brute-force search recovered the correct subkey. Although 1-round DES is algebraically trivial to invert given a single known plaintext-ciphertext pair, treating it as a target for differential cryptanalysis provided a concrete, code-level view of how biased differentials arise and how they can be exploited.

Our findings align qualitatively with the classical literature. Biham and Shamir's original work showed that full 16-round DES admits differential characteristics with non-negligible bias, but exploiting them requires about 2^{47} chosen plaintexts and substantial time, pushing such attacks beyond realistic deployment scenarios even though they are asymptotically faster than exhaustive key search. Matsui's linear cryptanalysis paints a similar picture from a different angle: linear approximations can be used to recover DES keys with around 2^{43}-2^{47} known plaintexts, again demonstrating structural weaknesses that are mostly of academic interest. Our 1-round experiments can be viewed as a scaled-down analogue of these results, where the same differential machinery becomes visible with far fewer pairs and the implementation details remain tractable for our cryptography class's final project.

At the same time, the limitations of our model are significant. We attack only the final round, not the full 16-round cipher; we assume ideal chosen-plaintext access to the encryption oracle; and we do not attempt to propagate differentials across multiple rounds or invert the full key schedule to recover the 56-bit master key. In practice, modern attackers would simply brute force single-DES or target Triple DES and legacy protocols, while contemporary standards have moved to AES and other ciphers with larger keys and stronger security margins. Our contribution is therefore best understood as a educational case study rather than a new break.

Nevertheless, the exercise is valuable. Implementing DES, building DDTs, and debugging the attack sharpened our understanding of how S-boxes, permutations, and key mixing interact under differential analysis. The fact that meaningful key-space reductions are achievable even in this simple setting reinforces why block cipher designers must select S-boxes and round counts with respect to differential and linear criteria, not just intuition. Future work could extend this framework to multiple rounds, compare DES's S-boxes against randomly generated alternatives, or replicate the experiment on modern lightweight ciphers to contrast their resistance profiles with DES.

Acknowledgments

Optional.

A Some supplementary material

Something you think is worth mentioning but not essential or digresses the flow of the main body, e.g., long proofs of some claims. Use it scarcely. (Is it really worth including? Unless 100% certain, the answer is most likely no.)