COMP 4109 Cryptography
Fall 2016 Project

Team Abdo

Name: Abdulbadi Sabir **Student Number:** 100907779
Name: Douglas Raymond **Student Number:** 100904379
Email: abdulbadisabir@cmail.carleton.ca

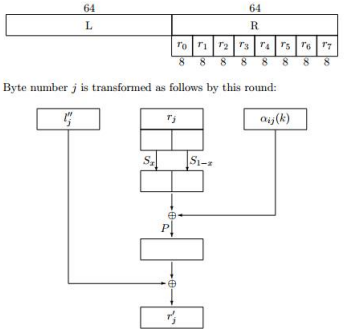
Introduction

The main objective of this paper is to compare and contrast the different types of attacks against the full 16 round DES cipher, and then to ultimately find one that breaks it faster than the exhaustive search's 2^{55} time. More specifically, this paper will explain and use differential cryptanalysis to break the cipher and show why the differential cryptanalysis technique is applicable to the DES and is superior to an exhaustive search.

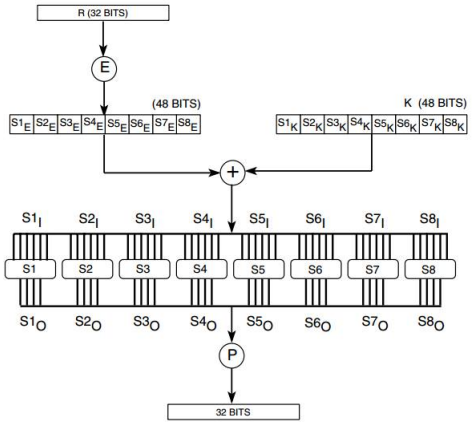
Iterated cryptosystems:

I*terated cryptosystems* are a family of cryptographically strong functions based on iterating a weaker function n times. An early *cipher* built upon the concept of iterated cryptosystems was called Lucifer, which uses a 48-bit key and operates on 48-bit blocks. The round functions in Lucifer uses two 4-bit S-boxes. The input is divided into groups of four consecutive bits and each group is translated by a reversible S box giving a four bit result, with each S-box getting permuted, inputting them into the following round. Decryption is carried out by going through the data from end to beginning, using the inverse of each S-box. DES was an enhanced version of Lucifer, and its functionality was based around the same concept of *iterated cryptosystems*.

The round function of Lucifer



The round function of DES



Meet in the middle attack:

The meet in the middle attack was introduced in 1987. It enabled the reduction of the key search for DES proportional to the number of rounds (the less rounds, the more the reduction). However, it was able to provide any reductions to the key search for a DES system which used 8 or more rounds. Their method included 4 steps. Let J be a set of data bits in a middle round with a set of keys bits I, where any in either I or J bits not affecting the other. Given I and J and a number of plaintext/ciphertext pairs the steps were as follows:

1. Try all the keys in which all the key bits in I are zero. Partially encrypt and decrypt a plaintext/ciphertext pair to get the data in the middle round. [3].

2. Get rid of the keys for which J bits are not the same under partial encryption/decryption[3].

3. For the remaining keys try all the possible values of the key bits in I[3].

The run time for this method was about $2^{56-|I|} + 2^{|I|}$, which is reasonable but not necessarily the fastest.

Exhaustive key search:
Diffie and Hellman proposed the exhaustive search of the entire keyspace on a parallel machine[3]. A Brute force attack is typically one where an attack tries as many passwords as possible with the intention to eventually arrive at the correct guess if there are a finite number of passwords[6].

For comparison three bruteforce attacks are referenced, for each the cost and time needed to crack the key have been mentioned[6].

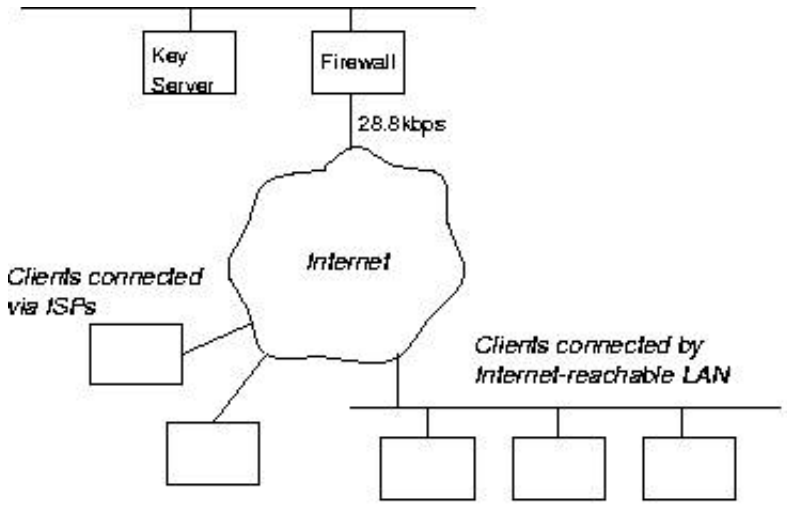
For our first example a known plaintext attack is considered as, generally in a commerical system, it would be impractical to require that the old plaintext be secret. In this scenario, the cryptanalyst will have several plaintext ciphertext pairs, all encrypted with the same key[6].

Let P and C be a known plaintext ciphertext pair with the same key K. For K = 1 to 2^{56} keys, decipher C until a correct key (K) is found that correctly yields the plaintext P[6].

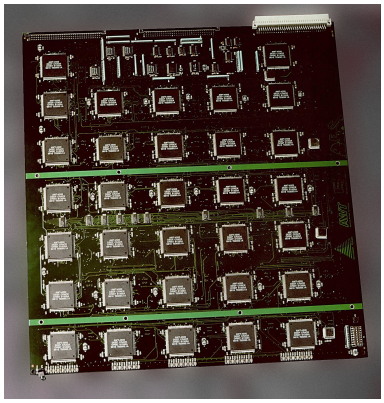
Although such a large iteration over a key space of $2^{56} = 10^{17}$ keys may seem infeasible, even if one key was tried every 10^{-6} seconds (microsecond), it would take 10^{11} seconds, or about 10^6 days to conduct an exhaustive search. A million devices searching in parallel would result in only one day for an exhaustive search and half a day of the average search, since the solution is usually found after trying on roughly half of the keys on average. In the attack, the idea was to develop a Large Scale Integration Chip that has a 10^6 Integrated Circuits (1,000,000 processors). Integrated Circuits (IC's) can work on low voltages, are very small in size, are cheap, can handle complex circuitry and considered a higher performance chip when compared to a discrete circuit alternative.[6]

As this cryptanalysis was done in 1977, the estimated cost proposed at the time will be several orders of magnitude cheaper today, due to advances in hardware and consequentially a decreased cost of hardware. When originally proposed, the machine's cost was estimated at 20$ million, after ten years however (1987) it was estimated to drop to 200,000$ a machine. For 2016, almost forty years later, we can expect the cost of such a machine, to be a fractional amount of 200,000$[6].

In 1997, the first DES Challenge was held, the DES Challenges are a series of brute force attack contests created by RSA security, designed to portray the lack of security provided by the Data Encryption Standard (DES) encryption scheme[7].



DESCHALL Project (DES Challenge) were the first team to solve the first challenge in 96 days. DESCHALL implemented a client server method with the idea of utilizing spare processing time on other machines connected to the internet to test for keys. A single 200 MHz Pentium computer could successfully test up to roughly one million keys/ second, if it was not doing any other processing. At that rate, it would have taken over 2,200 years to iterate through the entitre key space. However, the number of computers running the client rose quickly and at a point there were a maximum of over 14,000 unique hosts running DESCHALL in a day. By the time the correct key for the challenge was found, the algorithm had already covered a quarter of the keyspace and was searching roughly seven billions keys per second[7].



Following the first DES Challenge, at the DESChallenge II-2, Electronic Frontier Foundation's EFF DES Cracker cracked the key in less than three days. The DES Cracker is a machine that consists of a pc with a large array of "Deep Crack" chips, which were custom ASIC DES chips, one machine had over 1800 custom chips, spread out over 29 circuit boards which each consisted of 64 chips.

An observation of the EEF DES Cracker during DESChallenge II-2:

"Starting at 9:00 AM PST, Monday, July 13, 1998, the EFF DES Cracker began searching for the right key. The machine found the answer at 5:03 PM Pacific PST, Wednesday, July 15. Coincidentally, it took the EFF DES Cracker 56 hours to find a 56-bit key. When the EFF team started the search on Monday morning, they had 35868 search units running on 26 boards (each search unit examines 2.5 million keys per second). The team stopped the search for a few minutes on Tuesday night to improve the software and then again for a few minutes on Wednesday to add a 27th board, which sped up the machine slightly (to 37050 search units). The EFF DES Cracker searched 17,902,806,669,197,312 keys to find the correct answer, which averages out to a rate of 88,803,604,509 keys tested per second (88 billion). The machine was examining 92,625,000,000 keys per second when it found the answer. The key was found after searching almost exactly a quarter of the key space (24.8

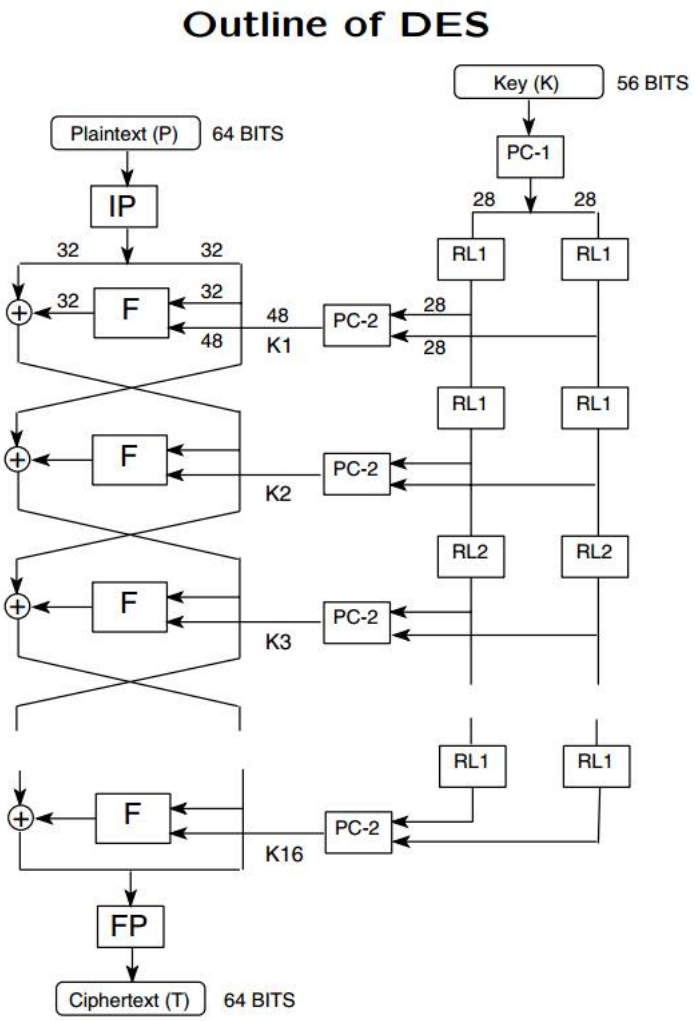
The machine cost an estimated $250,000 to build and decrypted the DES encrypted messaged from DESChallenge II-2 within only 56 hours of work. Half a year later, in 1999, EFF collaborated with distributed.net, using Deep Crack again to decrypt another DES encrypted message. However this time, the time to find the key was only about 22 hours.

Later that year, their success lead to a process of improvement and refining for the DES encryption scheme and DES was reaffirmed as a federal

standard, implemented as Triple DES however.

Differential cryptanalysis overview:

Differential cryptanalysis is a method that uses a plaintext attack for breaking certain classes of cryptosystems. Basically, it is the application of cryptanalysis to block and stream ciphers (and hash functions) in an effort to find a pattern relating the changes in input to the changes in output[2]. Differential cryptanalysis focuses on the occurrences of the plaintext and the differences into the last round of the cipher[4]. Let $[X = X_1\ X_2\ \mathbf{...}\ X_n]$ and output $[Y = Y_1\ Y_2\ \mathbf{...}\ Y_n]$. Also, let $X^{'}$ and $X^{''}$ be two inputs to the system with $Y^{'}$ and $Y^{''}$ being their respective outputs[4]. The difference in input is denoted by $\Delta\ X = X^{'} \oplus X^{''}$, therefore $\Delta\ X = [\Delta\ X_1\ \Delta\ X_2\ ...\ \Delta\ X_n]$ [4]. Likewise, $\Delta\ Y = Y^{'} \oplus Y^{''}$ represents the changes in output and $\Delta\ Y = [\Delta\ Y_1\ \Delta\ Y_2\ ...\ \Delta\ Y_n]$ [4].
In a purely randomized cipher, the probability that a difference in output ($\Delta\ Y$) occurs given a $\Delta\ X$ (difference in input) is $1/2^n$ where n represents the total number of bits in X [4]. $Differentialcryptanalysis$ seeks to find a $\Delta\ Y$ given a corresponding $\Delta\ X$ with a much higher probability than $1/2^n$, thus a pair ($\Delta\ X$, $\Delta\ Y$) is called a $differential$[4].



Let us contemplate the differential behaviour of an S-box, where there are 64^2 (4,096) possible pairs of input (X,X^*) (X and X^* are just arbitrary pairs of input) . While the 6-bit quantities X, X^* and X' are changing over their 64 possible values (where X' is X $\oplus$ X^*), the 4-bit quantities Y, Y^* and Y', (y = S(X), Y^* = S(X^*), and Y' = Y $\oplus$ Y^*) each diversify over their 16 possible value[2].

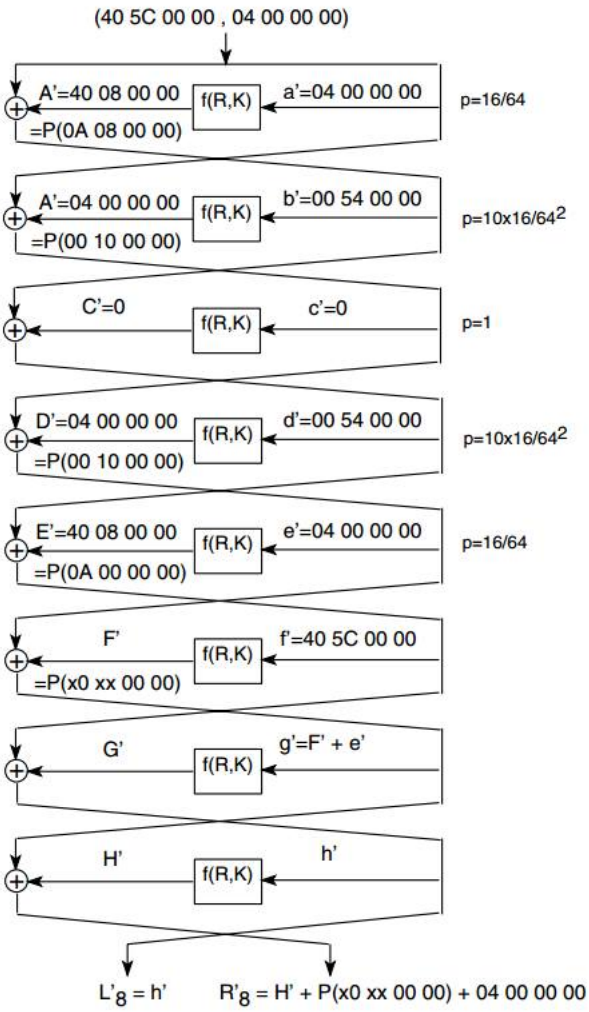
The distribution can then be calculated for the $differential$ output Y' for every one of the eight S-boxes by counting the number of times each value

Y' occurs as (X, X^*) alternates over its 4096 possible values[2].

DES with 8 rounds can be broken using a 5-round characteristic (observation of the occurrence per characteristic). The probability is calculated as follows"

$$\frac{16}{64} * \frac{10 * 16}{64^2} * \frac{16}{64} * \frac{10 * 16}{64^2} \approx 9.5 * 10^{-5}[2] \ .$$

We can assume that the characteristic occurs for every differential pair of plaintexts (X, X^*)[2].



We have a success rate of 1 in 10,000; thus we need several times this number of differential pairs. Biham and Shamir, who first introduced $differentialcryptanalysis$, assert that 150,000 pairs are needed[2].

Biham and Shamir proposed the following algorithm for breaking the 8 rounds DES cipher using $differentialcryptanalysis$, below is an example (depicted in the above diagram) using (40 5C 00 00 , 04 00 00 00) as input[2]:

Step 1: Find a pair (P, P^*) that have a differential P' that is equal to 40 5C

00 00 , 04 00 00 00 (where $P^{'}$ = P /*oplus* P^{*})[2].

Step 2: Obtain the ciphertext pair (C, C^{*} [2].

Step 3: Assume that the characteristic has occurred and obtain the differential outputs for S-boxes $S2$, $S5$, $S6$, $S7$ and $S8$ in the 8th round using $P^{-1}(H^{'}$ which is equal to $P^{-1}(R^{'}{}_{8} \oplus 04\ 00\ 00\ 00) \oplus (X0\ XX\ 00\ 00)$ [2]

Step 4: Test each of the 64 possible 6-bit subkeys $K_{8,2}$ associated with $S2$ in the 8th round to see which case the observed (X, X^{*}) to produce $Y^{'}$ obtained for S@ in Step 3 [2].

Place those subkeys that produce $Y^{'}$ in a table { $K_{8,2}$ }, repeating the above step to produce tables of the possible subkeys { $K_{8,5}$ }, { $K_{8,6}$ }, { $K_{8,7}$ }, and { $K_{8,8}$ } for $S5$, $S6$, $S7$, and $S8$, respectively[2].

Step 5: After determining the values of the tables in Step 4, look for any empty tables, and if one exists, then the characteristic could not have occurred, in which case, get rid of all 5 tables, go back to Step 1 and generate a new differential plaintext pair to work with[2].

Step 6: Since we're reached this step, we can assume that none of the tables are empty; that is to say that the characteristic has occurred.

In that case, generate all possible 30-bit portions of K_8 associated with Ss, $S5$, $S6$, $S7$, and $S8$ by choosing one 6-bit from each table[2].

If n_2, n_5, n_6, n_7, and n_8 denote the number of 6-bit subkeys in the tables { $K_{8,5}$ }, { $K_{8,6}$ }, { $K_{8,7}$ }, and { $K_{8,8}$ }, then N=$n_2n_5n_6n_7n_8$ would be the number of 30-bit values[2].

Let κ represent such a 30-bit value.

Step 7: For each of the κ's generated in Step 6, increment a counter corresponding to that value [2]

If any counter reaches a count of 1-, the associated κ value is probably correct[2]

If no cunter reaches 10, return to Step 1 and generate additional differential plaintext pairs[2].

The $differentialcryptanalysis$ attack described above is a kchosen plaintext attack, where the attacker has to obtain the ciphertext of any selected plaintext. It is also possible to carry out a known plaintext attack by allowing the attacker to pick from a larger set of plaintext/ciphertext pairs.

Given 2^{32} random plaintext/ciphertext pairs, suppose the chosen plaintext needs m pairs, there form $\frac{(2^{32}\sqrt{2m})^2}{2} = 2^{64}$m possible pairs of plaintexts [2].

The number of pairs creating each plaintext XOR value is about $\frac{2^{64}m}{2^{64}} = m$, since the bloack size is 64, there are 2^{64} possible plaintext XOR values. Conspicuously, with high probability there are about m pairs with each one of the several plaintext XOR values needed for $differentialcryptanalysis$.

Conclusion:

Table 1Comparison of Methods

	Diffie, W.; Hellman, M.E. (1977). "Exhaustive Cryptanalysis of the NBS Data Encryption Standard"	DESCHALL Project	EFF DES Cracker	EFF DES Cracker	Biham and Shamir (1990) "Differential Cryptanalysis"
Time taken to crack 56 bit DES Key	Theoretically one day with one million integrated circuits	96 Days	< 3 Days	<1 Day	To break the full 16 rounds, differential cryptanalysis requires 2^47 chosen plaintexts and will be completed in 2^37 time. (Varies)
Peak processing power (if applicable)		7 Billion keys per second	92,625,000,000 (92.6 billion) keys per second	239,295,116,728 (239.2 billion) keys per second	-
Year conducted	1977	1997	1998	1999	1990

With reference to table 1, we can see that as difficult as it may be to find a DES key, it is still possible to crack a 56 bit key given the correct resources and processing power. As the studies were conducted at different points in time, we are only comparing in terms of time efficiency (time taken to solve for key), the EFF DES Cracker(1999) ranks first. In terms of peak processing power the EFF DES Cracker again ranks first. The best method to crack the DES key would greatly depend on the resources available to the attacker. For a single standalone machine , the EFF DES Crack would rank the highest. For a client server attempt at finding the key, some variation of the DESCHALL Project could conceptually be faster. Considering that a full DES can be broken using 2^{47} chosen plaintexts in 2^{37} time regardless of if the keys are frequently changed, using Differential cryptanalysis, even though this method could take longer, it does seem like the most definite method in terms of obtaining a key from a changing set of keys.

Resources:

1. Eli Biham, Adi Shamir :Differential Cryptanalysis of the Full 16-round DES (2007)
   $http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.720.215\&rep=rep1\&type=pdf$

2. Cetin Kaya Koc :Differential Cryptanalysis
   $http://cs.ucsb.edu/\ koc/ccs130h/notes/dc1.pdf$

3. Eli Biham, Adi Shamir : Differential Cryptanalysis of DES- like Cryptosystems (1990)
   $http://www.cs.bilkent.edu.tr/\ selcuk/teaching/cs519/Biham-DC.pdf$

4. Howard M. Heys :A Tutorial on Linear and Differential Cryptanalysis
   $https://www.engr.mun.ca/\ howard/PAPERS/ldc_tutorial.pdf$

5. Eli Biham, Adi Shamir : Differential Cryptanalysis of the Data Encryption Standard
   $http://www.cs.technion.ac.il/\ biham/Reports/differential-cryptanalysis-of-the-data-encryptionstandard-biham-shamir-authors-latex-version.pdf$

6. Diffie, W.; Hellman, M.E. (1977). "Exhaustive Cryptanalysis of the NBS Data Encryption Standard".
   $https://www-ee.stanford.edu/\ hellman/publications/27.pdf$

7. Sabrina Schonhart, S. (2016). DES Challenge. [online] Cs-exhibitions.uni-klu.ac.at. Available at: $http://cs-exhibitions.uni-klu.ac.at/index.php?id=263$

8. Anon, (2016). [online] Available at:
   $https://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_des_$