Lecture 4: 27 January 2025

Instructors: Maria Francis and M. V. Panduranga Rao *Scribe: Gautam Singh*

The iterative characteristic by itself is not enough to break 16-round DES due to its low probability. However, it was enough for DES reduced to 15 rounds. To retain this probability, we use a new round 1. This new round 1 will generate plaintexts with XOR $(\psi, 0)$ which can then be fed into the characteristic. This gives rise to a modified 2R-attack which is fast enough to break DES. We describe the various stages of the attack below.

4.1 Data Collection Phase

We require to generate data whose XORed inputs into the iterative characteristic are $(\psi, 0)$. Suppose P is a 64-bit plaintext and let v_i be a 32-bit constant with the first 12 bits equal to the possible outputs of S1, S2, S3 after the first round and 0 elsewhere for $0 \leq i < 2^{12}$. Define for $0 \leq i < 2^{12}$

$$P_i = P \oplus (v_i, 0) \quad \bar{P}_i = (P \oplus (v_i, 0)) \oplus (0, \psi) \tag{4.1}$$

$$T_i = \text{DES}(P_i, K) \quad \bar{T}_i = \text{DES}(\bar{P}_i, K). \tag{4.2}$$

Then, $P_i \oplus \bar{P}_j = (v_k, \psi)$. Out of the 2^{24} possibilities of (i, j), each v_k occurs exactly 2^{12} times. Now, an XOR of ψ is fed into the first round, but we do not know which v_k is to be chosen initially to cancel the output of the F function and give us the desired $(\psi, 0)$ input to the second round.

However, trying all 2^{24} possibilities is slow. To find the right v_k, we exploit the cross-product structure of P_i and $\bar{P}_j$. Notice that a right pair will have zero outputs at S4, ..., S8 of the last round. Thus, we can feed in the plaintexts P_i and $\bar{P}_j$ to get outputs T_i and $\bar{T}_j$. These 2^{13} outputs can then be hashed by these 20 positions. A right pair will survive with probability 2^{-20}. Thus, out of the 2^{24} pairs, we will get only $2^4 = 16$ pairs.

We can further filter these pairs by testing against S boxes in rounds 1, 15 and 16. The input XOR values of S1, S2 and S3 in the first and the fifteenth rounds are fixed for right pairs. For the other boxes, we use the fact that about 80 % of the XOR pairs are possible. This reduces the number of surviving pairs to a fraction $(\frac{14}{16} \cdot \frac{13}{16} \cdot \frac{15}{16})^2 \cdot 0.8^8 = 0.0745$ of the original. Now, we are left with $16 \cdot 0.0745 = 1.19$ pairs per structure. Although right pairs are filtered, some wrong pairs may not have been filtered.

4.2 Data Analysis Phase

In previous attacks, data analysis entailed counting on certain subkey bits and then trying the most popular key values. This is memory intensive. In contrast, this attack uses negligible space by trying each suggested key value. A key value is suggested when it can create the output XOR of the last round and the expected output XOR of the first round using the particular plaintext and ciphertext pairs.

In the first and fifteenth rounds, the XOR inputs to S4, ..., S8 are all zero. The DES key scheduling algorithm ensures that all 28 key bits of the left register enters S1, S2 and S3 in the last two rounds and 24

key bits of the right register are used in the last round. Thus, $28 + 24 = 52$ bits enter the S boxes of the last two rounds. Considering there are $2^{32} \cdot 0.8^8$ possible outputs for the last round, there are only $X = \frac{2^{-32}}{0.8^8}$ fraction of keys left. Considering S1, S2 and S3 in each of the first and fifteenth rounds gives the factor $Y = \frac{2^{-12}}{\frac{14}{16} \cdot \frac{13}{16} \cdot \frac{15}{16}}$ for each round. Putting it all together, each pair suggests $2^{52} XY^2 = 0.84$ values of these 52 bits. Thus, each structure suggests $1.19 \cdot 0.84 \cdot 2^4 = 16$ choices for the whole key. To check whether a key is the right one, we can "peel" up the two additional rounds for each ciphertext pair and verify against against the characteristic, totaling to $16 \cdot 2 \cdot \frac{2}{16} = 4$ DES operations. The signal-to-noise ratio of this counting scheme is $S/N = \frac{2^{52} \cdot 2^{-47.2}}{\frac{1.19}{2^{12}} \cdot 0.84} = 2^{16.8}$.

We now describe the actual data analysis and extraction of the suggested keys. For the left register, we enumerate the 64 possibilities of $S4_{Kh}$, leaving on average 4 possibilities. From Figure 4.1, notice that three bits of $S4_{Kh}$ are shared with $S3_{Ka}$. Enumerating the other three bits reduces the average number of possibilities to two. Two bits of $S1_{Kh}$ are shared with $S3_{Ka}$. By completing the missing bits of $S1_{Kh}$ and then the two missing bits of $S2_{Ka}$, we can reduce the number of possibilities to about half on average. Completing the other 13 remaining bits of the left register in a similar way reduces the average number of suggested values of the left register to one. Similar counting methods are used on the right register to deduce 24 out of the 28 bits.

		K16									
		Left Key Register					Right Key Register				
		S1	S2	S3	S4	X	S5	S6	S7	S8	X
K1	S1		2	1	1	2					
	S2	2		1	2	1					
	S3	2			3	1					
	S4	2	3	1							
	X		1	3							
	S5							1	2	2	1
	S6						3		2	1	
	S7							2		2	2
	S8						2	3			1
	X						1		2	1	

Figure 4.1: Number of common key bits in K1 and K16.

4.3 Summary and Complexity

In summary, the following steps are performed to attack the entire 16 round DES.

1. Each structure contains a right pair with probability $2^{-35.2}$. The data collection phase uses 2^{35} structures, from which about $2^{35} \cdot 1.19 = 2^{35.25}$ pairs survive.
2. The probability that one of them is a right pair is about 58 %.
3. Analysis of the right pair leads to the correct key with high probability.

The time complexity of this attack is about $2^{35} \cdot 4 = 2^{37}$ equivalent DES operations.

A further reduction can be used by creating metastructures using the iterative characteristic with $\psi =$ `1B 60 00 00`$_x$. This metastructure would contain 2^{14} plaintexts.

We can parallelize the processing of each of these structures on up to 2^{33} processors with small local memories, one for each structure. Another advantage of this attack is that it can be done even when keys are changed frequently during data collection. The attack can be carried out incrementally with the number of available pairs, with success probabilities increasing with each new pair.

4.4 General Form of the Attack

The general form of this attack is stated in Theorem 4.1 without proof.

Theorem 4.1. *Given a characteristic with probability p and signal-to-noise ratio S/N for an iterated cryptosystem with k key bits, we can apply an attack which encrypts $\frac{2}{p}$ chosen plaintexts in the data collection phase and whose complexity is $\frac{2^k}{S/N}$ encryptions during the data analysis phase.*

Appropriately chosen metastructures can reduce the number of plaintexts to $\frac{1}{p}$. Further, the effective time complexity can be reduced by a factor of $f \leq 1$ if a wrong key can be discarded by carrying out a fraction f of the rounds.