CS 6160 Cryptology Lecture 8: Cryptanalysis of Block Ciphers

Maria Francis

September 22, 2020

Attacks on Block Ciphers

- We look at how to attack SPN and DES ciphers that have fewer than the lower bound rounds.
- We then look at the standard tool to attack block ciphers – differential attack.
- We give an overview of another standard tool, linear cryptanalysis.

Attacks on Reduced Rounds of SPN

- The strength of a block cipher depends on the number of rounds.
- The straightforward attacks on reduced rounds will help our understanding of these ciphers better.
- On one round with no final key-mixing step:
  - Adversary knows one input/output pair, $y = F_k(x)$.
  - He inverts y using public MP and S-boxes.
  - The intermediate that he has now is $x \oplus k$.
  - He knows x, so easily retrieves k.
- Shows how important key-mixing step is.

Attack on 1-round of SPN

- Full 1 round and key-mixing step.
- Assume a 64-bit block and a S-box with 8-bit input/output length.
- We assume two independent 64-bit sub-keys k_1, k_2 for the two key-mixing steps, master key, $k_1 \circ k_2$ is 128 bits long.
- First a simple idea: given a single input/output pair (x, y) launch a meet-in-the-middle like attack.
- For every possible k_2 invert the key-mixing step on y to get $y^{'}$.
- Now we have a full 1-round with no key-mixing step and we have seen how to get a unique k_1 corresponding to k_2 choice.
- Thus in 2^{64} time, we get 2^{64} possibilities for master key. More pairs and we get the result.

Attack on 1-round of SPN

- Another idea: Individual bits of the output depend on only part of the master key.
- Fix some input/output pair as before (x, y).
- The adversary will enumerate over all possible values for the first byte of k_2.
- XOR this value with y to get a candidate value for the output of the first S-box, invert it to obtain a cadidate for the input of that S-box.
- What is the input of the S-box? XOR of 8 bits of x and 8 bits of k_1 (the positions of these bits depend on the public MP)
- So now we have a candidate value for 8-bits of k_1.
- For each candidate for the first byte of k_2, there is a unique possible value for *some* 8 *bits of* k_1, the attack time is now reduced from 2^{16} to 2^8.

Attack on 1-round of SPN

- The attacker can tabulate all the feasible values in 2^8 time and for each byte he does this, giving him 8 lists.
- Number of possible master keys - $(2^8)^8 = 2^{64}$ values.
- The total time to do it – $8 \cdot 2^8 = 2^{11}$!
- Add more input/output pairs and you further reduce the space of possible keys : correct value of key should be consistent with more pairs and incorrect value can be consistent with new pair with probability no better than guessing, i.e. with 2^{-8}.
- The attack is possible because different parts of the key can be isolated from other parts. Way out: Diffusion with multiple rounds!

Attacks on Reduced Rounds of DES

Reading exercise: Attacks on Reduced-Round DES (in Section 6.2.3 in the Yehuda and Katz Textbook) Expect questions for your quiz from this section!

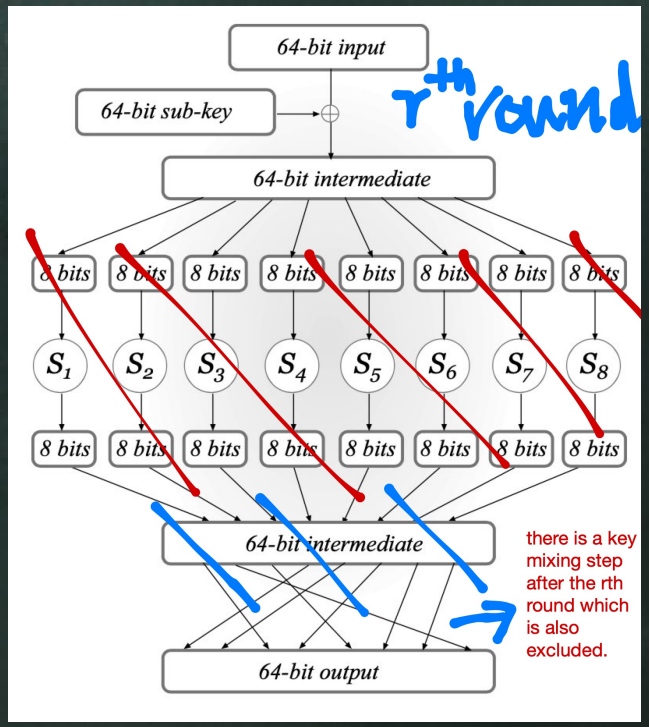
Differential Cryptanalysis

- A standard tool in cryptanalysis for block ciphers.
- Using this technique we can launch a CPA attack.
- Proposed first by Biham and Shamir to attack DES (1993).
- The idea:
  - We are not working with a random permutation, so can we obtain information regarding which specific differences in input (chosen plaintexts) will result in which specific differences in output with probability higher than a random permutation.
  - Consider a keyed permutation $F_k^{'}$ and for two uniform inputs x_1, x_2, let $\Delta_x = x_1 \oplus x_2$ and $\Delta_y = F_k^{'}(x_1) \oplus F_k^{'}(x_2)$.
  - What we are looking for is *differential (Δ_x, Δ_y) occurs in some keyed permutation $F_k^{'}$ with probability p that is higher than for a random permutation.*

Overview

- For any fixed (Δ_x, Δ_y) and x_1, x_2 s.t. $\Delta_x = x_1 \oplus x_2$, if we choose a uniform function $f : \{0,1\}^\ell \to \{0,1\}^\ell$, we have $Pr[f(x_1) \oplus f(x_2) = \Delta_y] = 2^{-\ell}$.
- Our way of attacking a weak block cipher : to look for differentials with significantly higher probability.
- The reason why they are useful is because we can use that to then launch a key-recovery attack.
- We show that for SPNs. Let F be a block cipher with ℓ-bit block length and a r-round SPN.
- Let $F'_k(x)$ be the intermediate result after doing the key mixing step of round r.
- I.e, *we exclude the S-box step, mixing permutation of the last round and the final $r+1$ key-mixing step.*

What we are looking at?



How to get the final sub-key?

- Let there be a (Δ_x, Δ_y) in F' that occurs in F' with $p >> 2^{-\ell}$. We can retrieve the final mixing sub-key k_{r+1}.
- $\{(x_1^i, x_2^i)\}_{i=1}^{L}$ – a collection of L pairs of random inputs s.t. $\Delta_x = x_1^i \oplus x_2^i \ \forall i$.
- Consider a chosen-plaintext attack: Obtain the encryption of x_1^i, x_2^i for all i as ${y_1}^i = F_k(x_1^i)$, ${y_2}^i = F_k(x_2^i)$.
- For all possible bit-strings $k^* \in \{0,1\}^\ell$, compute ${y_1}^{i*}, {y_2}^{i*}$ as outputs of $F'(x_1^i), F'(x_2^i)$ along with the S-box substitution, mixing permutation and XORing with the final sub-key as k^*.

How to get the final sub-key?

- If $k^* = k_{r+1}$ then a p-fraction of the pairs will satisfy ${y_1}^{i^*} \oplus {y_2}^{i^*} = \Delta_y$.
- Else heuristically only a $2^{-\ell}$ fractions will yield this differential.
- For a large enough L the correct value for the final sub-key k_{r+1} can be determined.
- But we have to try out 2^{ℓ} possibilities, guess portions of k_{r+1} at a time.
  - Assume S-boxes in F have 1-byte input/output and focus on the first byte of Δ_y.
  - We verify if the differential holds in the first byte by guessing only 8-bits of k_{r+1}, the 8-bit output of the first S-box.
  - Try out all 2^8 possibilities and see which one yields the desired differential with the highest probability ($\approx p + 2^{-8}$) and that can be used to guess that portion. Why $p + 2^{-8}$ and not p?

An Example

4-round SPN, $\ell = 16$ bits, S-box with 4-bit input/output length.

Input:	0000	0001	0010	0011	0100	0101	0110	0111
Output:	0000	1011	0101	0001	0110	1000	1101	0100
Input:	1000	1001	1010	1011	1100	1101	1110	1111
Output:	1111	0111	0010	1100	1001	0011	1110	1010

Mixing Permutation :

In:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Out:	7	2	3	8	12	5	11	9	10	1	14	13	4	6	16	15

Finding a differential

- Consider a differential $\Delta_x = 1111$.
- $S(0000) \oplus S(1111) = 0000 \oplus 1010 = 1010$ a difference of 1111 in inputs yields a difference of 1010 in the outputs.
- We now need to see if this differential happens very often.
- For e.g: $S(0001) \oplus S(1110 = 0001 \oplus 1111) = 1110$ is a counterexample.
- But here is another one that also gives the same differential: $S(0100) \oplus S(1011) = 1010$.

Finding a differential

Below figure shows this differential happens with probability 1/2.

x	S(x)	x ⊕ 1111	S(x ⊕ 1111)	S(x) ⊕ S(x ⊕ 1111)
0000	0000	1111	1010	**1010**
0001	1011	1110	1110	0101
0010	0101	1101	0011	0110
0011	0001	1100	1001	1000
0100	0110	1011	1100	**1010**
0101	1000	1010	0010	**1010**
0110	1101	1001	0111	**1010**
0111	0100	1000	1111	1011
1000	1111	0111	0100	1011
1001	0111	0110	1101	**1010**
1010	0010	0101	1000	**1010**
1011	1100	0100	0110	**1010**
1100	1001	0011	0001	1000
1101	0011	0010	0101	0110
1110	1110	0001	1011	0101
1111	1010	0000	0000	**1010**

All possible differentials

- Carry this process for all 2^4 input differences Δ_x to calculate the probability of the differential.
- For each pair (Δ_x, Δ_y), we make a table of the number of 4-bit input x s.t.

$$S(x) \oplus S(x \oplus \Delta_x) = \Delta_y.$$

Differentials in our S-box

- (i, j) – no of inputs with difference i that map to outputs with difference j. 8 pairs for $(1111, 1010)$. $(0100, 0110)$ has also high probability of $6/16 = 3/8$.

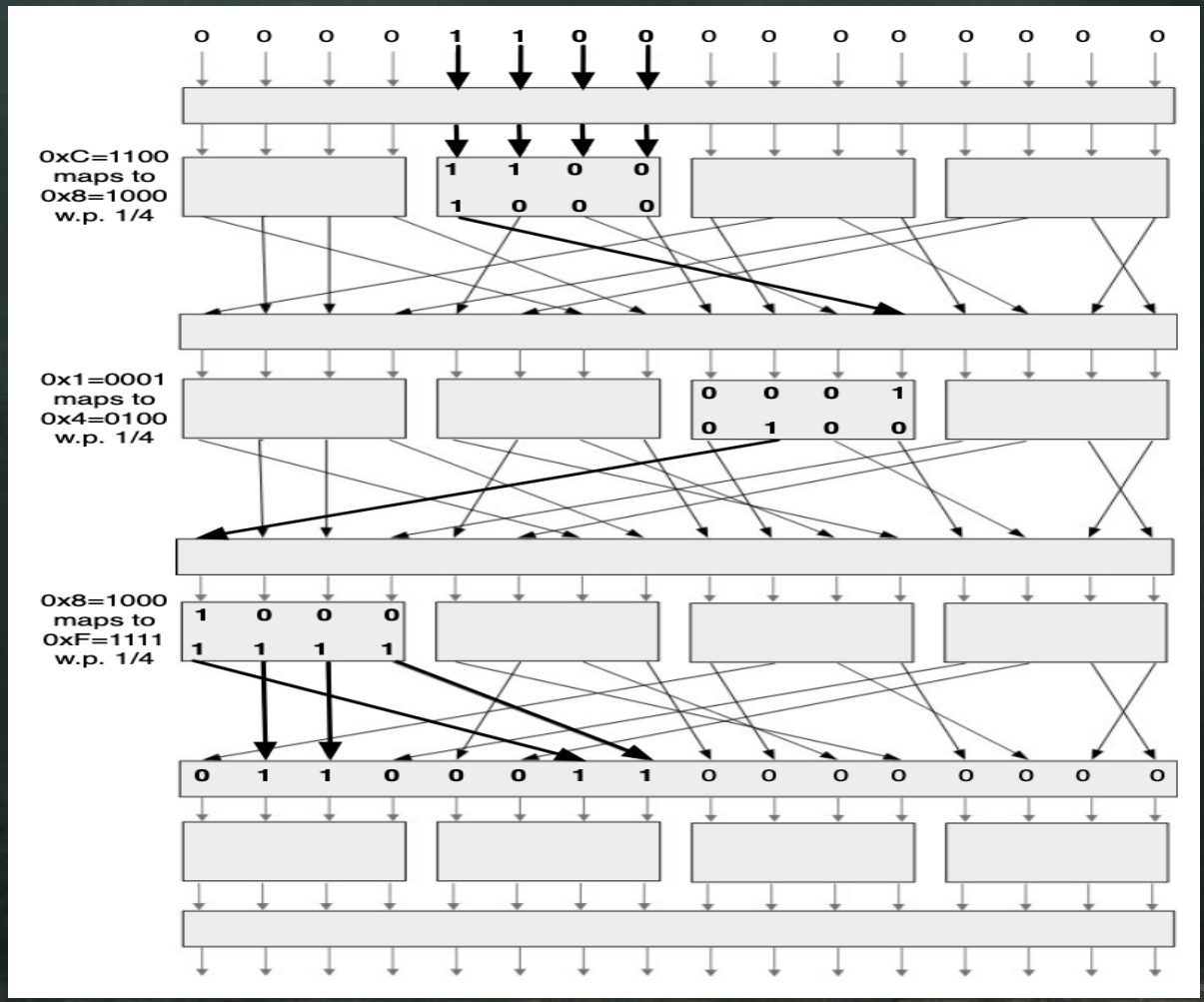
Output Difference Δ_y

Input Difference Δ_x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	4	0	0	0	2	2	2	2	0	0	4	0
2	0	0	0	0	0	2	0	2	0	2	2	4	2	2	0	0
3	0	2	2	4	0	4	0	0	0	0	0	0	0	2	2	0
4	0	0	0	2	2	2	6	0	2	0	0	0	2	0	0	0
5	0	2	2	0	0	0	0	0	4	0	0	0	4	2	2	0
6	0	2	0	2	0	0	0	0	0	2	0	2	0	4	0	4
7	0	2	0	0	2	4	2	2	0	2	0	0	0	2	0	0
8	0	0	0	2	0	0	0	2	0	0	0	2	2	2	2	4
9	0	2	0	2	2	2	0	4	0	2	2	0	0	0	0	0
A	0	0	4	0	2	0	2	4	2	0	2	0	0	0	0	0
B	0	0	2	0	0	0	2	0	0	2	0	0	4	2	4	0
C	0	0	0	0	0	0	0	0	4	4	0	4	0	0	0	4
D	0	4	2	2	0	0	2	2	0	0	0	0	0	0	0	4
E	0	2	4	2	4	0	0	0	0	0	0	0	2	0	2	0
F	0	0	0	0	0	2	2	0	2	0	8	2	0	0	0	0

Differentials for our S-box

4-round SPN

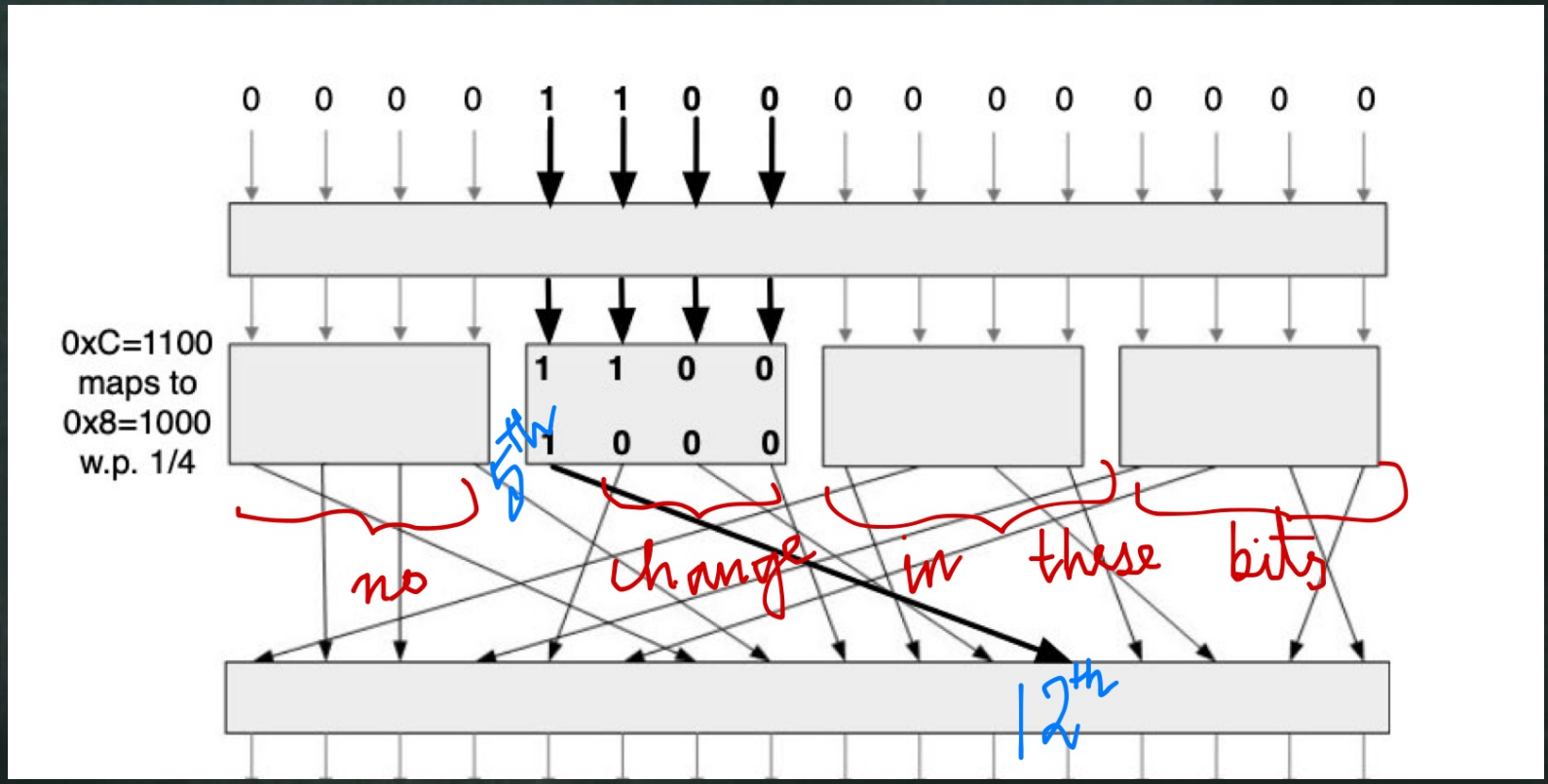
Consider two 16-bit inputs that have a differential 0000 1100 0000 0000.



4-round SPN

- Key mixing step is just XOR - no impact on differential.
- Inputs to the second S-box in the first round have the differential 1100.
- From our previous table, *Fig. Differentials for our S-box*, a difference in 1100 yields a difference of 1000 with probability 1/4.
- With probability 1/4 the differential in the output of the 2nd S-box after round 1 is a single bit which is moved from the 5th position to 12th position by the mixing permutation.
- The inputs to other S-boxes are equal and so outputs are equal and differential is 0000.

After one round



After 4 rounds

- So now for the third S-box, input difference of 0001 gives an output difference of 0100 with prob. $1/4$.
- Once again one output bit difference is moved to 1st position from 10th position by MP.
- In the third round there is an output difference of 1111 with probability of $1/4$ and the bits $1, 2, 3, 4$ are moved to $7, 2, 3, 8$.
- Overall, $\Delta_x = 0000\ 1100\ 0000\ 0000$ results in an output difference $\Delta_y = 0110\ 0011\ 0000\ 0000$ after three rounds with probability $\frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{64}$.
- For a random function it is $2^{-16} = 1/65536$ so the differential here happens with significantly higher probability than a random function.

Recovering final sub-key k_5

- $\{(x_1^i, x_2^i)\}_{i=1}^{L}$ - L pairs of random inputs with Δ_x differential.
- Using a CPA, we obtain the values $y_1^i = F_k(x_1^i)$ and $y_2^i = F_k(x_2^i)$ for all i.
- For all possible initial 8 bits of k_5 we compute the initial 8 bits of y_1^{i*}, y_2^{i*}, the intermediate value after the key mixing step of the 4th round. All that we need is invert S-boxes and MP.
- Now if we had used the correct 8 bits of k_5 then the 8 bit differential 0110 0011 would have occurred with at least 1/64 probability.
- For a random incorrect key it would have been $2^{-8} = 1/256$ probability.
- With more pairs (large L) we can retrieve the key!

Differential Attacks in practice

- Famous one: Differential attack broke FEAL-8, an alternative to DES with just 1000 chosen plaintexts.
- Any proposed cipher should be tested for resistance to differential cryptanalysis.
- But on DES, only slightly better than brute-force search, needs 2^{47} chosen plaintexts.
- Most real-world applications you cannot get that many pairs.
- Small changes to S-boxes can make DES vulnerable to differential attacks.

Linear Cryptanalysis

- Developed by Matsui in the early 1990s.
- Basic idea: Identify linear relationships between input and output that occur more than a random function.
- Formally, we define a linear bias ϵ: Bit positions $i_1, \ldots, i_{in}$ and $i'_1, \ldots, i'_{out}$ have linear bais ϵ if for uniform x and k and $y := F_k(x)$, it holds that

$$|Pr[x_{i_1} \oplus \cdots \oplus x_{i_{in}} \oplus y_{i'_1} \oplus \cdots \oplus y_{i'_{out}}] - \frac{1}{2}| = \epsilon$$

  where x_i, y_i denote ith bit of x and y.
- For a random function we expect the bias to be close to 0. Just an XOR of bit values should give 0 or 1 with probability $\frac{1}{2}$.

Linear Cryptanalysis & Cipher Design

- Matsui showed if you have a large enough bias in a cipher then you can retrieve the secret key.
- This attack only needs known plaintexts! It is much easier to obtain a huge encrypted file of known plaintext than gathering encryptions of plaintexts of our choice.
- Attack on DES: 2^{43} known plaintext/ciphertext pairs.

Design of Block Ciphers:

- Design S-boxes with minimal differential probabilities and linear biases.
- Cannot eliminate all since some differential will happen more frequently than others.
- Increasing rounds helps.

Security of DES

- The best known attack is still exhaustive key search.
- In 1997 DES was broken in 96 days and by next year 14 days and then in just 56 hours.
- The last attack was broken by a special machine called Deep Crack built at a cost of a quarter of a million.
- The short block length is also a concern.
- For e.g: we will soon see that an attacker can break the security of the encryption scheme with prob. $2q^2/2^\ell$ if it has q plaintext/ciphertext pairs.
- For DES that means only $q = 2^{30}$ plaintext/ciphertext pairs are needed for security to be compromised with high probability.
- But the design is still perfect.