First exam hints Cryptanalysis Part

4b Compute the total success probability over all round 1&2 differentials and round 3&4 differentials with given ΔP and ΔC

For any given round 1&2 differential D_{12} and round 3&4 differential D_{34} with probabilities p_{12} and p_{34}, the success probability for the boomerang is $p_{12}^2 \cdot p_{34}^2$.
Hence one needs to sum $p_{12}^2 \cdot p_{34}^2$ over all possible D_{12} and D_{34} under the constraints:

$$\sum_{D_{12}\ with\ \Delta P}\ \sum_{D_{34}\ with\ \Delta C} P[D_{12}]^2 \cdot P[D_{34}]^2$$

As the lecture notes say as these choices are independent we can rewrite it as a product of sums:

$$\left(\sum_{D_{12}\ with\ \Delta P} P[D_{12}]^2\right) \cdot \left(\sum_{D_{34}\ with\ \Delta C} P[D_{34}]^2\right)$$

And we can determine both sums independently.
If we focus on the first sum.
And if we condition one the first round Sbox differential F7 (1111 => 0111) (having probability 2/16=1/8, see DDT) then we have three active round 2 sboxes.
For each of the three active round 2 sboxes we need to pick a differential (with input difference 4 as in the hint): let's call them A, B, C.
Then the contribution of this choice to the total success probability over round 1&2 is:

$$P[D_{12}]^2 = \left(\frac{1}{8}\right)^2 \cdot P[A]^2 \cdot P[B]^2 \cdot P[C]^2$$

We need to sum over all choices A,B,C:

$$\sum_A \sum_B \sum_C \left(\frac{1}{8}\right)^2 \cdot P[A]^2 \cdot P[B]^2 \cdot P[C]^2$$

As those choices are independent we rewrite:

$$\left(\frac{1}{8}\right)^2 \cdot \left(\sum_A P[A]^2\right) \cdot \left(\sum_B P[B]^2\right) \cdot \left(\sum_C P[C]^2\right)$$

Note that all three sums are the same term from the hint in the exam: $2 \times \left(\frac{4}{16}\right)^2 + 4 \times \left(\frac{2}{16}\right)^2$
That can be determined from the sum of the squares of the probabilities one finds in row '4' of the DDT.
Now one needs to do the same for the other choices for the first round Sbox differential.
And of course the same for round 3&4, but then backwards.

4c Prove that all such 3-round differentials are impossible.
Note that having only 1 active Sbox in round 1 means active round 2 sboxes have input difference 1,2,4 or 8.
Note that having only 1 active Sbox in round 3 means active round 2 sboxes have output difference 1,2,4 or 8.
Note that all differentials {1,2,4,8}x{1,2,4,8} in the DDT have probability 0.

6a Fill tables
For e.g., '..-' one needs to evaluate the following boolean function outcomes:
(x,y,z)=(001) vs (x',y',z')=(000),
(011) vs (010)
(101) vs (100)
(111) vs (110)

(the first two positions are constant for either side, but can be 0 or 1, the third position is on the left hand side 0 and on the right hand side 1:
In short z=1, z'=0 as $\delta z =$ z'-z = -1, x'=x, y'=y as $\delta x = \delta y = 0.$)
Convention is to subtract right hand side outcome from left hand side outcome: f(x',y',z')-f(x,y,z).

E.g., then one finds that having zero output difference is impossible, thus column g=0 is n/a.
To obtain g={+1} the first two positions need to be different, thus that column is '!.-'.
To obtain g={-1} the first twe positions need to be the same, thus that column is '^.-'.

6b determine path over steps 48,...,63:
Use tables from 6a for $\Delta F_i[21]$:
\Delta F_62 : +.. => choose 0: +.0 so \Delta F_62 = 0
\Delta F_63 : ++. => choose 0: ++1 so \Delta F_63 = 0
The rest follows from tracking a single bit difference through these equations:
- \delta T_t = \delta Q_t-3 + \delta W_t + \delta F_t
- \delta R_t = RL(\delta T_t, RC_t)
- \delta Q_t+1 = \delta Q_t + \delta R_t

6c determine path over steps 32,...,47 (backwards)
Similar like 6b for $\Delta F_i[21]$ (use 6a), but use equations backwards:
- \delta R_t = \delta Q_t+1 - \delta Q_t
- \delta T_t = RL(\delta R_t, 32-RC_t)
- \delta Q_t-3 = \delta T_t - \delta W_t - \delta F_t

6d Prove that one can use up to r near-collisions to obtain a collision
note that (0,2^i,2^i,2^i) for i=0,..,31 is a basis for the space (0,x,x,x) with x\in Z/2^32 Z

6e Show how you can obtain minimal number of near-collisions
use BSDR NAF, zie lecture notes

6f write down an algorithm that obtains $\delta CV = (0, x, x, x)$

Apply birthday search with distinguished points
use f: {0,1}^96 -> {0,1}^96: (x,y,z) -> (a,b-c,b-d)
where (a,b,c,d)=compress(CV_i-1, ...|x|y|z) if say x mod 2 = 0
where (a,b,c,d)=compress(CV'_i-1, ...|x|y|z) if say x mod 2 = 1