Luke Pepin

[Aidan Mcdonald](mailto:Aidan.McDonald@uconn.edu)

CSE 3400 Problem Set 2

3/4/24

1. Let G be a PRG that stretches the input by two (i.e., G : {0, 1} n → {0, 1} 2n). Consider the following function and show it is not a PRF. Do this by constructing a distinguisher and computing P(DF k (·) = 1) and P(Df (·) = 1) and showing the difference between the two probabilities is large. The function F takes an n bit key and a 2n bit input x and is defined as follows:

Fk(x) = x ⊕ G(xᴸ ⊕ xᴿ ⊕ k) ⊕ 0101 · · · 01

where xL are the left n bits of x and xR are the right n bits of x.

Consider distinguisher D:

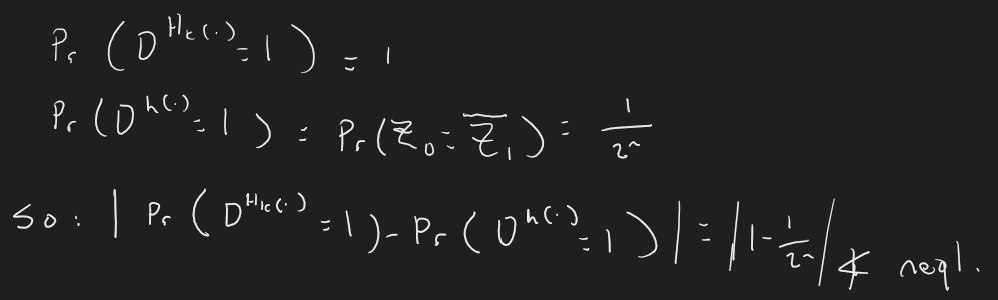
Query x = 0 … 0, recv. Z₀

Query x = 1 … 1, recv. Z₁

If Z₀ = Z₁ ⊕ 1 … 1, output 1

Else: output 0

Probability:



1. Let Fk be a PRF. Consider the following function F 0 : F 0 k (x) = Fk(x) if x is even Fk(x ⊕ 0 · · · 01) if x is odd, Show that F 0 is not a PRF even though F is. As with the previous problem, construct a distinguisher and work out the two probability values.

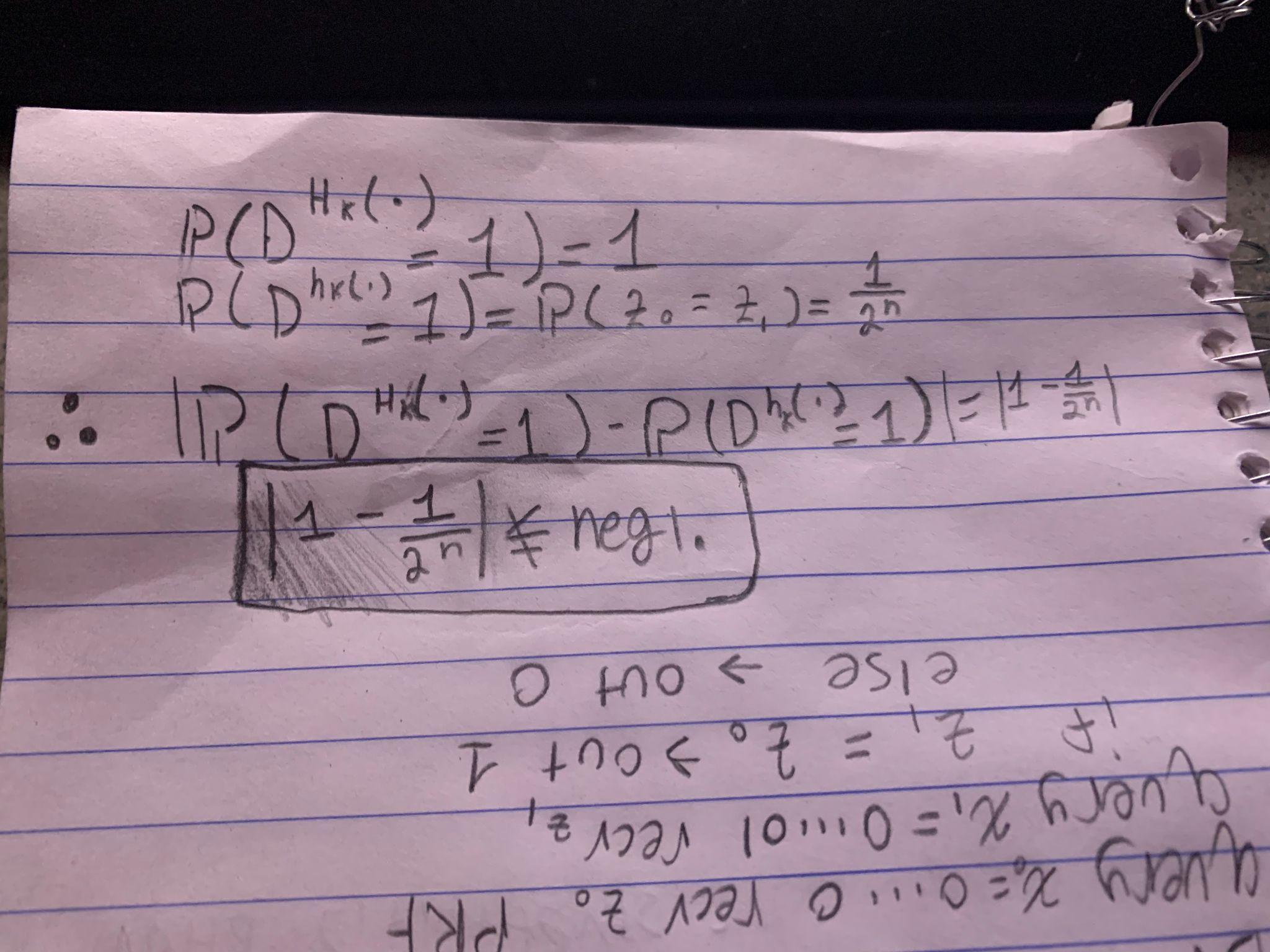
Consider distinguisher D:

Query x0=(0…0), recv Z0 = (Fk(0…0))

Query x1=(0…01), recv Z1 = (Fk(0…01⊕0…01)) = (Fk(0…0))

If Z1 = Z0 out 1,

Else: out 0



1. (Two parts): Let Fk be a PRP and consider the following encryption scheme for messages of size 2n:

• Gen draws an n bit key uniformly at random

• Enc(k, m1m2) = Fk(m1) || Fk(m2 ⊕ 1 · · · 1)

Note that we break the 2n bit message m into two parts m = m1m2 where each mi is n bits.

**Problem Part 1**: Construct the Dec function.

Given key k & C = Fk(m1) || Fk(m2 ⊕ (1…1))

Dec(key k, ciphertext C) function can be defined as follows:

1. m1 = Fk-1(C[0:n-1])
2. m2 = Fk-1(C[n:2n-1] ⊕ (1…1))
3. Return: m = m1 || m2

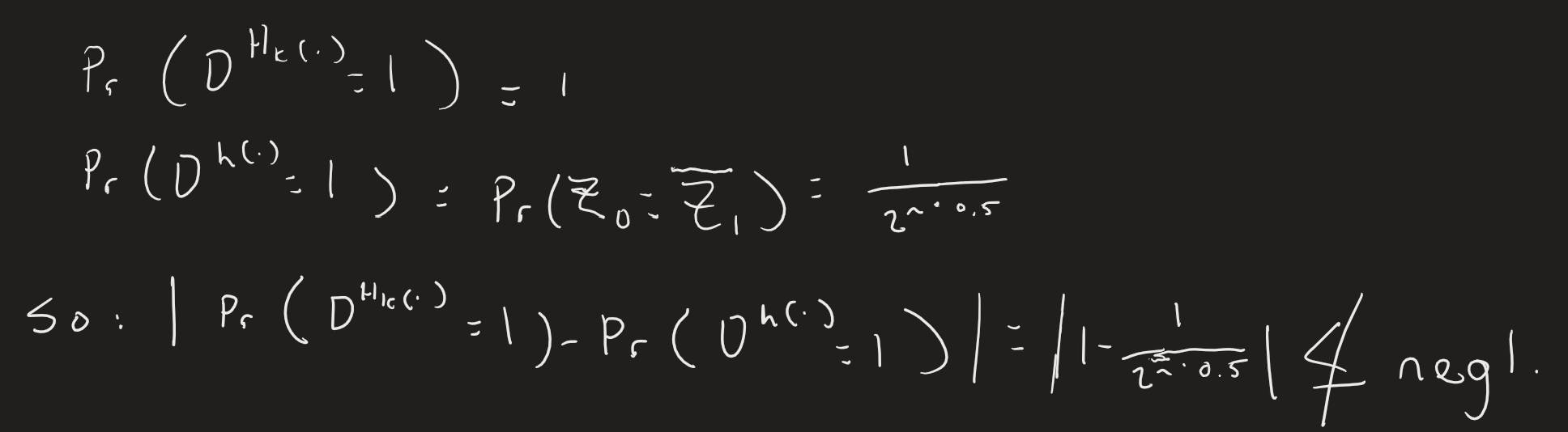
**Problem Part 2**: Show that this is not EAV secure.

Consider distinguisher D:

Query x0=(0…0), recv (R0,L1)

If L0= R1 ⊕ (1…1) output 1

Else: output 0



1. Consider a two-round Feistel Network where each round uses the same PRF and same key. Show that this is not a PRF even if the component function is a PRF (refer to Section 2.6.3 and especially Figure 2.21 in the text which shows three-rounds).

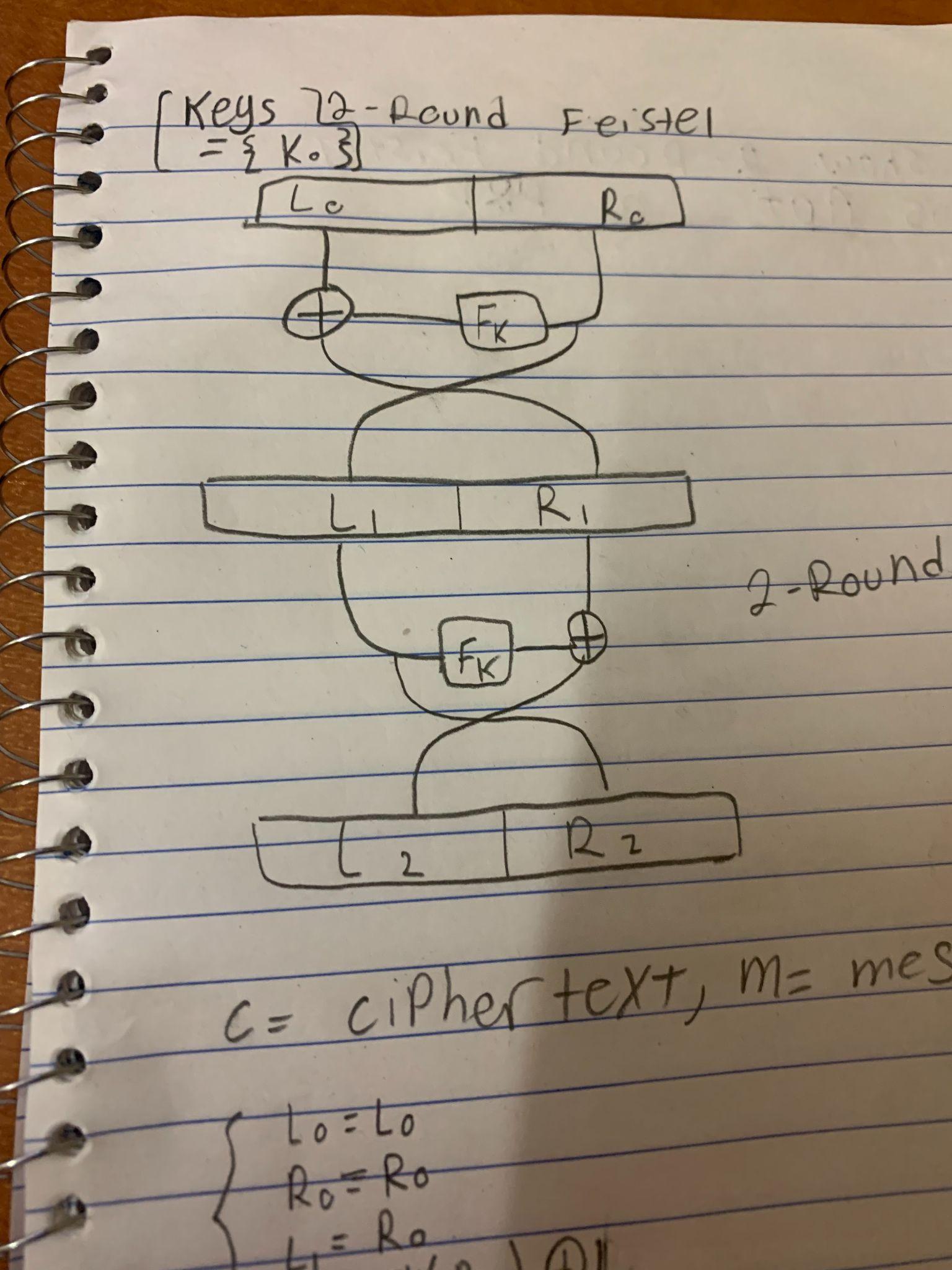
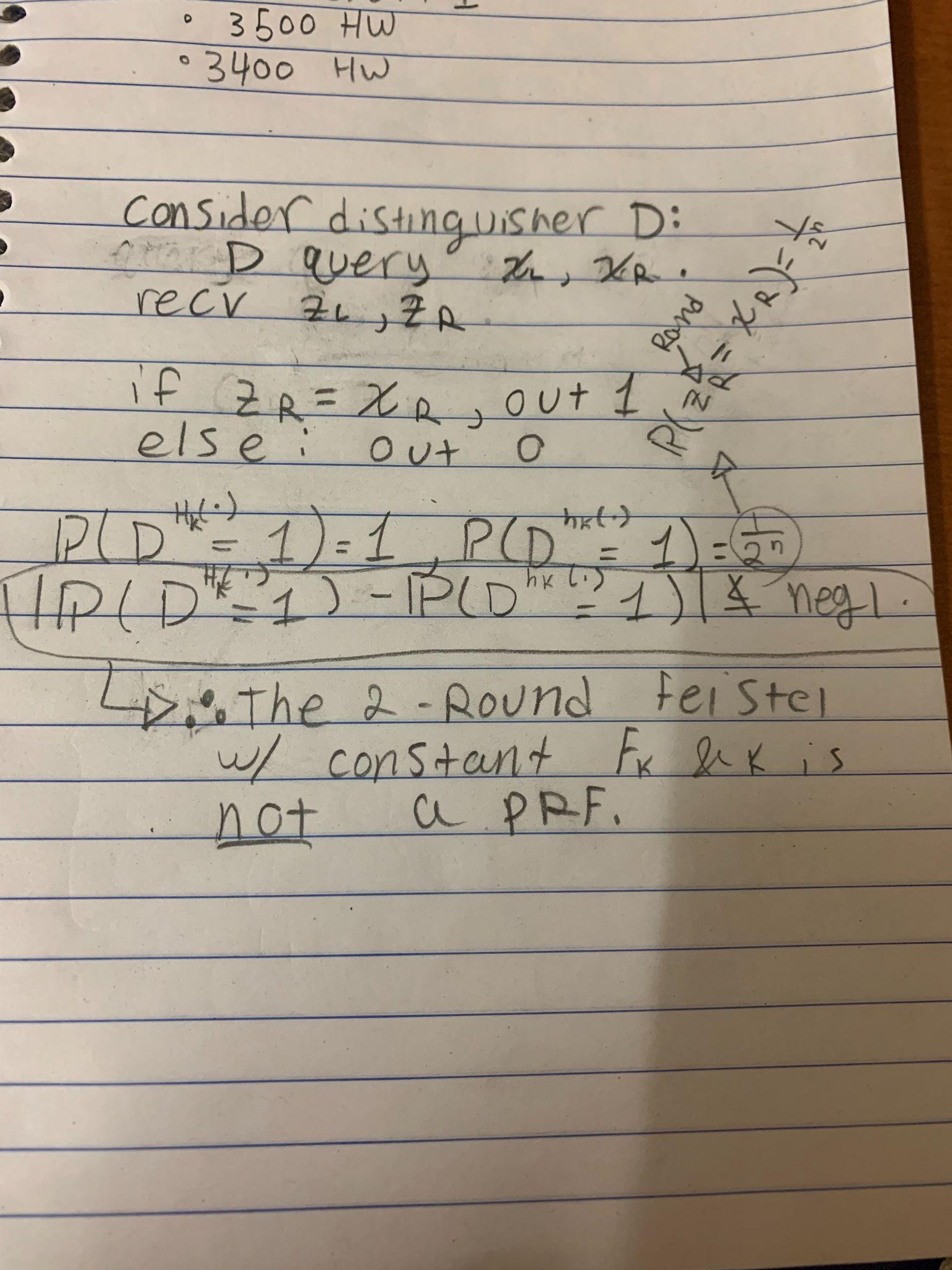
Consider the following 2-round Feistel Network where each Fk call uses the same key and same PRF F.

Figure of network →

To show that this is not a PRF, let's analyze each round and what their left and right output is.

| Round | Left output | Right output |
| --- | --- | --- |
| 0 (begin) | L0 = L0 | R0 = R0 |
| 1st round | L1 = R0 | R1 = Fk(R0) ⊕ L0 |
| 2nd round (end) | L2 = Fk(L1) ⊕ Fk(R0) ⊕ L0  ~⇒ L2 = Fk(R0) ⊕ Fk(R0) ⊕ L0 | R2 = L1 = R0 |

We see that the right output at the end of the second round of the Feisten network is the same as the input in the first right input, or alternately ‘R0’ = ‘R2’. Based on that fact we can create a distinguisher.

1. Similar to the above, but now consider a three-round Feistel Network where each round uses the same PRF and same key. Show that this is not a PRF even if the component function is a PRF (refer to Section 2.6.3 and especially Figure 2.21 in the text which shows three-rounds). Note that if the key is different each round, then this is a PRF! So, in your proof, assume the key is identical for all three rounds. HINT: First write out the algebra for what the output is in terms of Fk. Then, make a first query. Your second query should be a function of the first response. That is, your next query must use the response from the first query in some manner.

| Round | Left Output | Right Output |
| --- | --- | --- |
| 0 (begin) | L0 = L0 | R0 = R0 |
| 1st | L1 = R0 | R1 = Fk(R0) ⊕ L0 |
| 2nd | L2 = R1 | R2 = Fk(R1) ⊕ L1 |
| 3rd | L3 = R2 | R3 = Fk(R2) ⊕ L2 |

Consider distinguisher D:

Query (R0,L0) =(0…0), recv (R1,L1)

Query (R1,L1), recv (R2,L2)

If R1=L2 output 1

Else: output 0