3.6

a) G'(s) = G($s_1 s_2 \dots s_n$), $s = s_1 \dots s_{2n}$

Assume for contradiction that there exists a PPT distinguisher D' and a function negl such that : $| \, P[D'(G'(s_{2n})) = 1] - P[D(r) = 1] \, | > negl(n)$, r$\in \{0,1\}^{l'(n)}$

Where $l'(2n) = l(n) > 2n$

Then we construct a PPT distinguisher D for the generator G. Upon input $r \in \{0,1\}^{l(n)}$ distinguisher D invokes distinguisher D' upon input r and outputs whatever D' outputs.

We have that $P[D(G(s_n)) = 1] = P[D'(G'(s_{2n})) = 1]$

And $P[D(r) = 1] = P[D'(r) = 1]$

Therefore $| \, P[D(G(s_n)) = 1] - P[D(r) = 1] \, | > negl(n)$

This is a contradiction with the pseudorandomness of G, so G' is pseudo random.

b) G'(s)=G($0^{|s|}||s$), s=$s_1 \dots s_{n/2}$

G' not pseudorandom.

Take a pseudogenerator G" with an expansion factor $l''(n) = 2l(n) > 4n$ and define G(s) = G($s_1 s_2$) $= G''(s_1)$ where $2|s_2| = 2|s_1| = |s|$. We know from (a) that G must be a pseudorandom generator with expansion factor $l(n) > 2n$.

Now for any s, $G'(s) = G(0^{|s|}||s) = G''(\,0^{|s|})$. Thus G' can be distinguished from a random string by computing $G''(\,0^{|s|})$ and comparint it to the input.

c) $G'(s) = G(s)||G(s + 1 \, mod \, 2^n)$

G' is not pseudorandom.

First let's show that $G''(s) = G(s_1 \dots s_{|s|-1})||s_{|s|}$ is pseudorandom by contraditcion. Assume G" is not then there exists a distinguisher D" which succeeds with non negligeable probability. Let's construct a distinguisher D for G. On input r, D outputs whatever D" outputs on r||b where b is a random bit. If r=G(s), then $G(s)||b = G''(s||b)$. So D can succeed with a non negligeable probability. This contradicts the pseudorandomness of G therefore G" is pseudorandom.

Now assume G' pseudorandom. Then G''(s)||G"($s + 1 \, mod \, 2^n$) is also pseudorandom.

Take n=2, s=0000.

Then $G''(0000)||G''(0000 + 1 \, mod 2^1) = G(0000)||G(0001) = G(000)||0||G(000)||1$

A distinguisher on input $r \in \{0,1\}^4$ just has to check whether $r_1 = r_3$ and can succeed with non negliegeable probability.

3.13

FA,b : {0, 1} → {0, 1} by FA,b(x) = Ax + b

Let's construct a distinguisher D that succeeds with a non negligeable probability.

D obtains b by querying F(0). Now D can obtain A by querying F for all unit vectors. Now D can easily distinguish a true random r from F(x) with the inverse of A and -b (NB: the inverse exists as it's needed for decryption), so F is not pseudorandom.

3.18

•Dec : F is a pseudorandom function so there exists F^{-1}. Compute $F^{-1}(c) = r||m$ then retrieve the second half that is corresponding to m (both r and m have same length so it's easy to know where the concatenation happens).

•Assume this scheme is not CPA-secure then there exists an adversary A who can recover information about the messages from the ciphertexts. This implies that A can distinguish between a truly random string and Fk(r||m). This is a contradiction with the pseudorandomness of F, therefore the scheme is CPA-secure.

•Consider F is a strong pseudorandom permutation. Assume this scheme is not CCA-secure. Then there exists an adversary A who can recover information about the ciphertexts from the messages (it cannot be the inverse since we already know the scheme is CPA-secure). This implies that A can distinguish between a truly random string and Fk(r||m) using the inverse of F. This is a contradiction with the strong pseudorandomness of F, therefore the scheme is CCA-secure.

3.19
a) This scheme doesn't have indistinguishable encryptions since an eavdropper can obtain G(r), on input $< r, G(r)\oplus m >$, and then compute G(r) $\oplus(G(r)\oplus m) = m$. Therefore the scheme cannot be CPA-secure neither.

b) This scheme Π has indistinguishable encryptions.
Assume Π doesn't have indistinguishable encryptions. From the proof of theorem 3.18 we know it implies that there exists an adversary A who can distinguish between Fk(0^n) from a random string, but this contradicts the pseudorandomness of F. Therefore scheme Π has indistinguishable encryptions.
However this scheme isn't CPA-secure as encryption is deterministic.

c) This scheme is CPA secure and thus has indistinguishable encryptions.
Indeed this scheme is using CTR mode with 2 blocks with a modification :
the blocks are computed with $ri\oplus mi$ where ri = Fk(r+i-1) , instead of computing the ith block as $ri\oplus mi$ with ri = Fk(r+i). This modification does not alter the CPA-security since Fk(r) is indistinguishable from a true random string as F is pseudorandom.

3.17
Consider F' a pseudorandom permutation such that $F'k(k) = 0^{|k|}$.
Compute $yi = F'^{-1}(0^i)$ for i = 1,2,... and for each compute $F'(yi)$, stop when $F'(yi) = O^{|yi|}$
This yi is the key. Therefore a distiguisher with access to F' and the inverse of F' can distinguish between F' and a random permutation. F' is not a strong permutation.

3.29
Generate a random $k \in \{0,1\}^n$ for $m \in$ M $with\ |m| = n$
Enc : <Enc1(k), Enc2($m\oplus k$)> = <c1 , c2>
Dec : Dec1(c1) ⊕ Dec2(c2)

The two ciphertexts are independent of m since they're based on a random string, so an arbirtary adversary cannot learn anything about m without both decryptions. Indeed if only scheme2 is CPA secure A can obtain k but it's unuseful with c2, conversely if A can obtain $m\oplus k$,without having k A cannot learn anything on m (1-time pad).
Therefore the scheme is CPA-secure.

5.5
a) Outputting all 0s at each round function just means that the the xor at each round changes nothing. Therefore the output of the Feistel network is (L0,R0) if r is even and (R0,L0) if r is odd since L0 and R0 switch place at each round.

b) If the round function is the identity function then at each round the xor consists $Li\oplus Ri$ if the inputs whet Li and Ri at the previous round. However XORing the same element cancel

the xor so Li and Ri will always be either L0,R0 or $L0 \oplus R0$. In addition the feistel network switches places of the input so we get as possible outputs:
at round $i \equiv 0[3]$, the output is (L0,R0)
at round $i \equiv 1[3]$, the output is (R0, $L0 \oplus R0$)
at round $i \equiv 1[3]$, the output is ($L0 \oplus R0, L0$)

5.6

Let f be the DES mangler function. For every i we have :

$$f(ki, R) = E(R) \oplus ki$$

And $f(\overline{ki}, \bar{R}) = E(\bar{R}) \oplus \overline{ki}$
$= \overline{E(R)} \oplus \overline{ki}$ because E() simply duplicates half of the bits
$= E(R) \oplus ki$ by XOR construction
$= f(ki, R)$

Therefore the S-boxes will output the same result since they are deterministic.
Then by noticing that all permutations in DES have the property $P(\bar{R}) = \overline{P(R)}$, we can conclude that the final step : applying a mixing permutation will output such that:
$DES_k(x) = \overline{DES_k(\bar{x})}$

5.12

a) Given (m1,m2). In $O(2^n)$ find all k1 such that $m1 = F_{k1}(m2)$. Then in O(c) find all the k2 such that $m2 = F_{k2}^{-1}(m1)$. Store the keys found. To increase the probability of success repeat this process with 2 other pairs of inputs. Take the intersection of the 3 sets of keys found and with high probability it will be the correct keys.

b) If we computed beforehand for every k2, $m = F_{k2}^{-1}(0^n)$ and we stored the k2,m's sorted then it would be possible to find for which k2 we have $m2 = F_{k2}(0^n)$ in a constant amount of time.

c) Take x such that $F_{k1}(x) = 0^n$ (we know k1 and F_{k1}^{-1} must exist for decryption), then look up in our stored table from Qb for which keys k2 we have $x = F_{k2}^{-1}(0^n)$. This can also be done in a constant amount of time. Then for each of those keys k2 compute $F'_{k1,k2}(x)$ and check if it is equal to y. If it is then this k2 is the correct key.

d) Do the preprocessing of Qb, and run for every k1 as described in Qc. Request the encryption of x each time to be able to do the last step.

Homemade Exercises

1. $\pi_{k1,k2}(x_1, x_2) := \langle x_1 \oplus F_{k1}(x_2), x_2 \oplus F_{k2}(x_1 \oplus F_{k1}(x_2))\rangle$

Compute $\pi_{k1,k2}(x_1, x_2) = \langle l1, r1 \rangle$

Notice $l1 \oplus l1 = (x_1 \oplus F_{k1}(x_2)) \oplus (x'_1 \oplus F_{k1}(x_2))$
$= (x_1 \oplus x_1)$
$=0$

Therefore it is easy for a distinguisher to recognize with a non negligeable probability if <w,w'> is a true random or not by checking if $w \oplus w' = 0$

2. $\pi_{k1,k2,k3}^{-1}(0,0) = \langle l, r \rangle = \langle F_2(F_3(0)), F_3(0) \oplus F_1(0) \rangle$
$\pi_{k1,k2,k3}(0, l) = \langle l1, r1 \rangle = \langle l \oplus F_2(F_1(l)), F_1(l) \oplus F_1(l1) \rangle$

$\pi_{k1,k2,k3}^{-1}(r1 \oplus r, l1) =< l2, r2 > = < l \oplus l1 \quad , \ldots >$

By checking if $l2 = l \oplus l1$ an adversary can know if it is truly random or not. Therefore $\pi_{k1,k2,k3}$ is not a strong pseudorandom permutation family.

3. Construct Π' the same as Π, since the key is not public we will keep CPA-security but by setting pk=sk it will not have indistinguishable encryptions.