Review

Material: everything until padding oracle attacks (inclusive)

OTP, secret sharing, negligible, birthday bounds

PRG, PRF, PRP, CPA, modes, malleability, padding oracle

1. **[10 points; 2 per part]** True/false

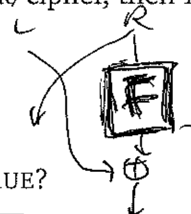
   T F It is possible for a deterministic encryption scheme to be CPA-secure, if it is based on a PRP instead of a PRF. Enc same thing twice ⇒ same ctxt

   T F If an encryption scheme has CPA$ security then it also has CPA security.
   actual theorem in book

   T F Suppose an adversary sees a ciphertext $c = \text{Enc}(k, m)$ and then later learns what m was. If the scheme is CPA secure, then the adversary cannot solve for k.
   if you could solve for k, ask for CHALLENGE(mL, mR) → c → Decrypt using k ⇒ break CPA

   T F $\frac{1}{\sqrt{2^n}}$ is negligible.
   $= \frac{1}{2^{n/2}}$

   T F If F is used as the *round function* of a Feistel network/cipher, then F must be invertible.



this always invertible

not necessarily invertible

2. **[10 points; 5 per part]** Short answer:

   (a) What is the probability that this program outputs TRUE?

```
FOO():
  x ← {0,1}^λ
  y ← {0,1}^λ
  z ← {0,1}^λ
  return y =? x ⊕ z
```

breakpoint

fixed

Pr[random = random]
$= \boxed{\frac{1}{2^\lambda}}$

only 1 value of z makes this true ⇒ $\boxed{\frac{1}{2^\lambda}}$

   (b) In a t-out-of-n Shamir secret sharing scheme, what should be the degree of the polynomial?

$\boxed{t-1}$

$$p(x) = \underline{p_0} + \underline{p_1} x + \underline{p_2} x^2 + \cdots \underline{p_{t-1}} x^{t-1}$$

t coefficients

line = deg 1 ⇒ 2 pts
parabola = deg 2 ⇒ 3 pts
⋮
1 $\boxed{t-1}$ ⇒ t pts

3. **[20 points; 10 per part]** Medium-length answers

(a) Suppose you have access to a secure PRF $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ (not necessarily a secure PRP). Describe any CPA-secure way of encrypting λ-bit plaintexts.
For full points:

- Describe both the encryption & decryption algorithm. Note that decryption cannot assume that the PRF has an inverse.
- You do **not** have ~~to give a security proof.~~

$$\boxed{\begin{array}{l} \underline{Enc(k,m):} \\ \quad r \leftarrow \{0,1\}^\lambda \\ ret(r,\ F(k,r) \oplus m) \end{array}}$$

$\star$ malleable

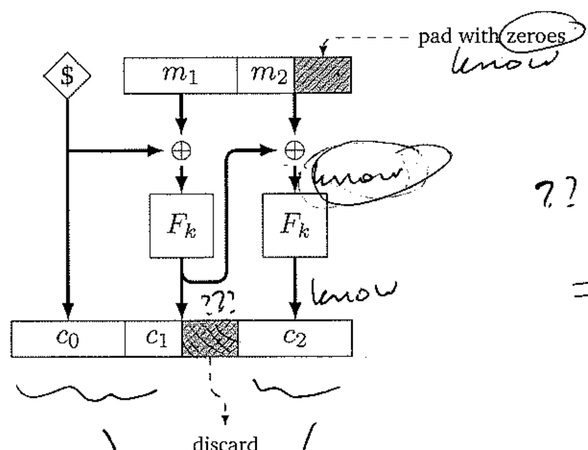
given (x,y)

$\Downarrow$

$Dec(x,y) \oplus s$

$= Dec(x, y \oplus s)$

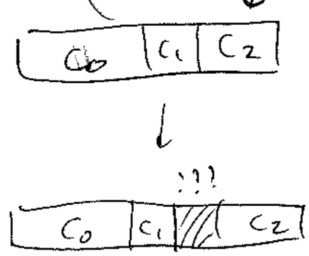
(b) Consider using ciphertext stealing with CBC mode on a 16-byte block cipher. The plaintext is 24 bytes (so 1.5 blocks long). Ciphertext stealing says to pad the plaintext with 8 bytes of zeroes, encrypt with CBC mode, and throw away the last 8 bytes of the middle ciphertext block.
Explain how the receiver recovers the missing 8 bytes.



?? ⊕ 000... = known

⇒ ?? = known

given



find missing bits of c_1

$\boxed{\text{last 8 bytes of } F^{-1}(k, c_2)}$

4. **[20 points]** Let F be a secure PRP with blocklength λ. Consider the following encryption scheme:

$$\boxed{\begin{array}{l}\underline{\mathsf{Enc}(k,m):}\\ \quad r \leftarrow \{0,1\}^\lambda \\ \quad x := F(k,r) \\ \quad y := F(k,r) \oplus m \\ \quad \text{return } (x,y)\end{array}}$$

Show that the scheme does not satisfy CPA security (I will also accept if you show that the scheme does not satisfy CPA$ security). **For full points**, explicitly write the distinguisher (as a program that ~~uses the~~ appropriate libraries' interface), and compute its output probabilities in the presence of the two relevant libraries.

obs: $x \oplus y = [F(k,r)] \oplus [F(k,r) \oplus m]$

$= m$

obs: $\mathrm{Enc}(k, 0^\lambda)$ has the form (x,x)

Attack:

CPA attack

$$\boxed{\begin{array}{l}(x,y) = \text{CHALLENGE}(0^\lambda, 1^\lambda)\\ \text{return } x \stackrel{?}{=} y\end{array}}$$

Pr[output true in left library] = 1

since (x,y) is Enc of 0^λ

Pr[output true in right lib] = 0

since $x = y \oplus 1^\lambda = \overline{y}$

$\Rightarrow x \neq y$

CPA$ attack

$$\boxed{\begin{array}{l}(x, y) = \text{CHALLENGE}(0^\lambda) \\ \text{return } x \stackrel{?}{=} y\end{array}}$$

$\Pr[$ output true in "real" lib $] = 1$

(actual Enc of 0^λ)

$\Pr[$ output true in "rand" lib $] = \frac{1}{2^\lambda}$

5. **[20 points]** Suppose $F : \{0,1\}^\lambda \times \{0,1\}^{\text{in}} \to \{0,1\}^{\text{out}}$ is a secure PRF. Based on F we define the following functions:

$x \to \{0,1\}^{in}$ $\qquad\qquad$ $in = out$

$H(k,x)$: return $F(k,x) \| F(k,\overline{x})$	$H(k,x)$: return $F(k,x) \| F(k,F(k,x))$
(students enrolled in CS427)	(students enrolled in CS519)

Here "$\|$" means concatenation and "$\overline{x}$" means the bitwise complement of x (flip every bit).

Show that H **is not a secure PRF.** For full points, explicitly write the distinguisher (as a program that uses the appropriate libraries' interface), and compute its output probabilities in the presence of the two relevant libraries.

$L \| R = \text{QUERY}(0^{in})$

$L' \| R' = \text{QUERY}(1^{in})$

return $L \stackrel{?}{=} R'$

can say:
pick arbitrary x
QUERY(x)
QUERY$(\overline{x})$

in real library

$L = F(k, 0^\lambda)$

$R' = F(k, \overline{1^\lambda})$

$= F(k, 0^\lambda)$

so $\Pr[\text{output true}] = 1$

in rand library

$L \leftarrow \{0,1\}^{out}$

$R' \leftarrow \{0,1\}^{out}$

$\Rightarrow \Pr[\text{output } \cancel{\text{two}} \text{ true}] = \frac{1}{2^{out}}$

$$\boxed{\begin{array}{l} L \| R = \text{QUERY}(0^{in}) \\ L' \| R' = \text{QUERY}(L) \\ \text{return } R \stackrel{?}{=} L' \end{array}}$$

in real lib:

$L = F(k, 0^{in})$

$$\text{equal} \left\{ \begin{array}{l} R = F(k, \underline{F(k, 0^{in})}) \\ \quad = F(k, L) \\ L' = F(k, L) \end{array} \right.$$