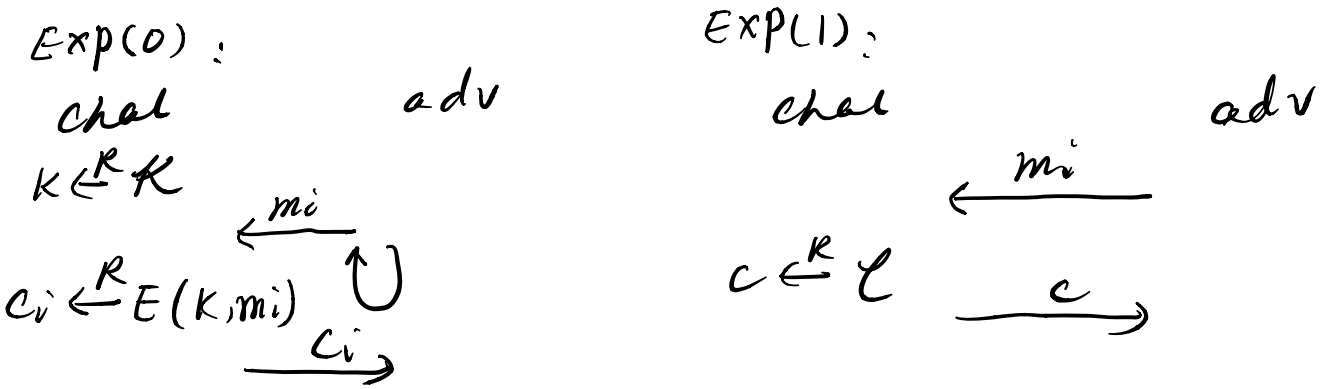
5.7 (pseudo-random ciphertext security). In Exercise 3.4, we developed a notion of security called pseudo-random ciphertext security. This notion naturally extends to multiple ciphertexts. For a cipher $\mathcal{E} = (E, D)$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, we define two experiments: in Experiment 0 the challenger first picks a random key $k \xleftarrow{R} \mathcal{K}$ and then the adversary submits a sequence of queries, where the ith query is a message $m_i \in \mathcal{M}$, to which the challenger responds with $E(k, m_i)$. Experiment 1 is the same as Experiment 0 except that the challenger responds to the adversary's queries with random, independent elements of $\mathcal{C}$. We say that $\mathcal{E}$ is pseudo-random multi-ciphertext secure if no efficient adversary can distinguish between these two experiments with a non-negligible advantage.

(a) Consider the counter-mode construction in Section 5.4.2, based on a PRF F defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, but with a fixed-length plaintext space $\mathcal{Y}^\ell$ and a corresponding fixed-length ciphertext space $\mathcal{X} \times \mathcal{Y}^\ell$. Under the assumptions that F is a secure PRF, $|\mathcal{X}|$ is super-poly, and ℓ is poly-bounded, show that this cipher is pseudo-random multi-ciphertext secure.

(b) Consider the CBC construction Section 5.4.3, based on a block cipher $\mathcal{E} = (E, D)$ defined over $(\mathcal{K}, \mathcal{X})$, but with a fixed-length plaintext space $\mathcal{X}^\ell$ and corresponding fixed-length ciphertext space $\mathcal{X}^{\ell+1}$. Under the assumptions that $\mathcal{E}$ is a secure block cipher, $|\mathcal{X}|$ is super-poly, and ℓ is poly-bounded, show that this cipher is pseudo-random multi-ciphertext secure.

(c) Show that a pseudo-random multi-ciphertext secure cipher is also CPA secure.

(d) Give an example of a CPA secure cipher that is not pseudo-random multi-ciphertext secure.



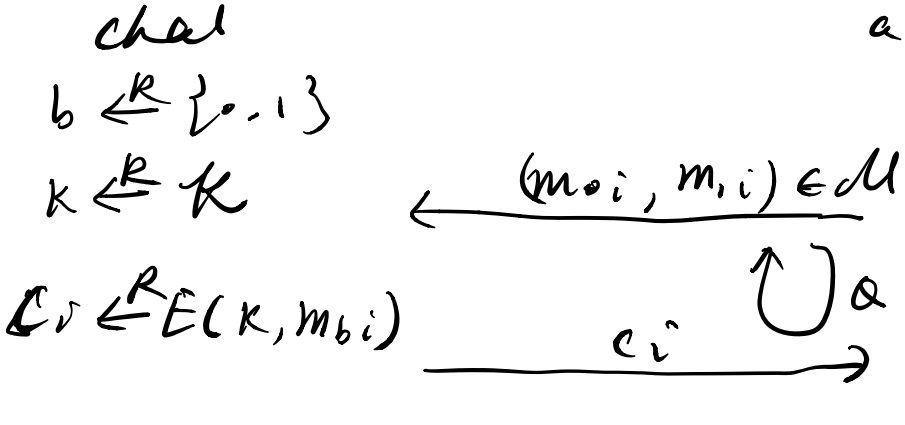
PRMCS adv [A, $\mathcal{E}$]

PRMCS ⟶ CPA security

CPA security ⟶ PRMCS

intuition:
in Exp(1),
random ciphertext
is harder than
$E(k, m_{bi})$

CPA game (bit-guessing)

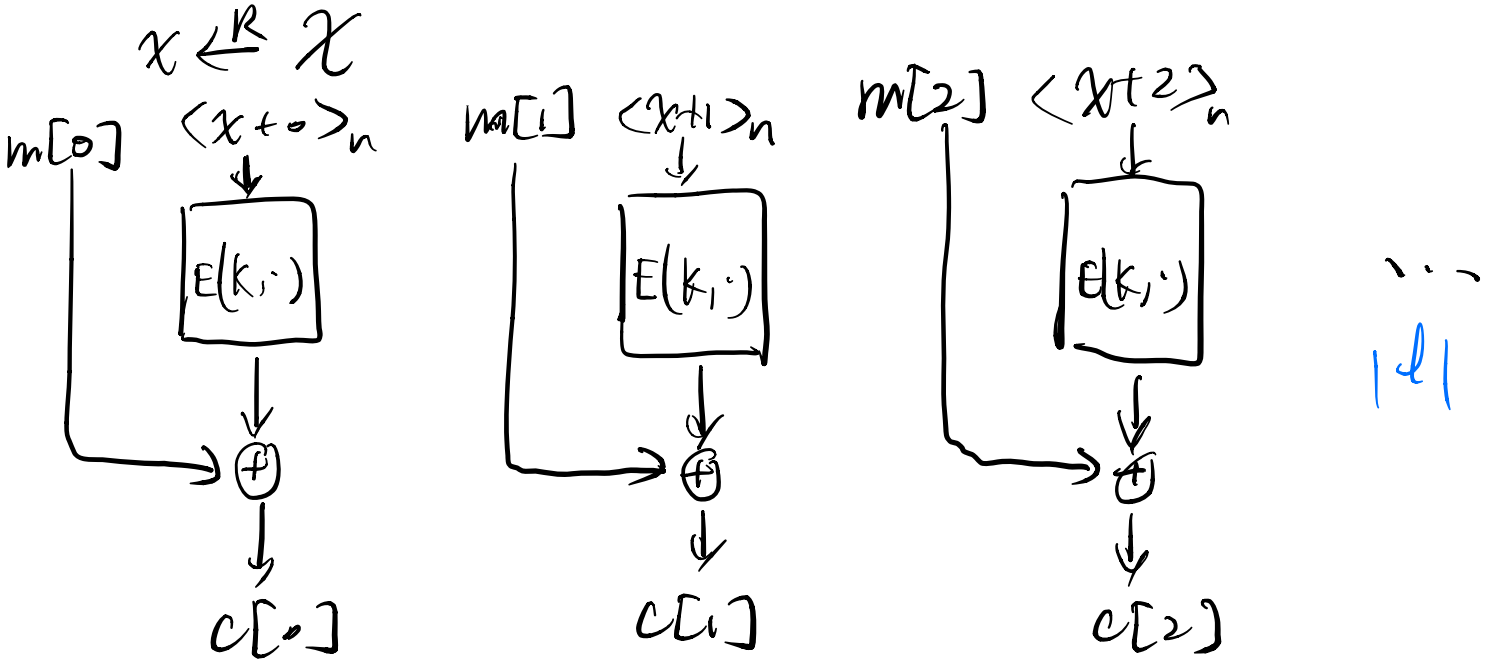


$b \in \{0,1\}$

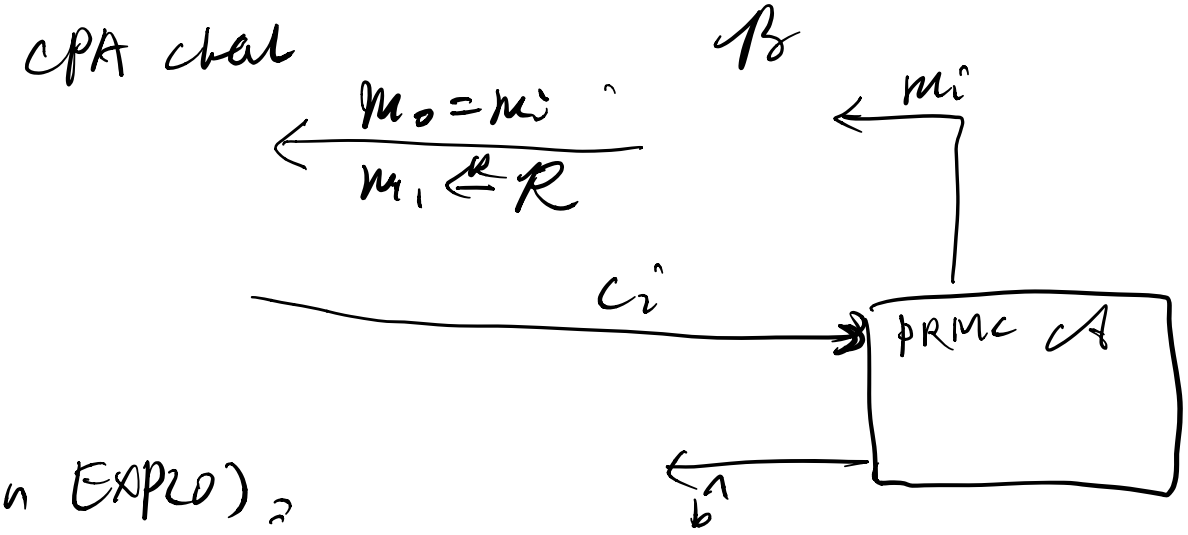
$$CPA^{\$}adv[A,\mathcal{E}] = \left| Pr[W_0] - Pr[W_1] \right|$$

* why "chosen plaintext"? : if $m_{0,i} = m_{1,i}$, then effectively it's the same as PRMC game in Exp(0).

(a) ctr-mode:



Prove: CPA $\longrightarrow$ PRMC in ran-ctr mode.



in EXP(0),

$Pr[W_0] = p_0$

← prob. of PRMC adv outputting 1 in its Exp(0) game.

$$\Pr[W_1] = P_1 + \ell \cdot SSadv[A, E]$$

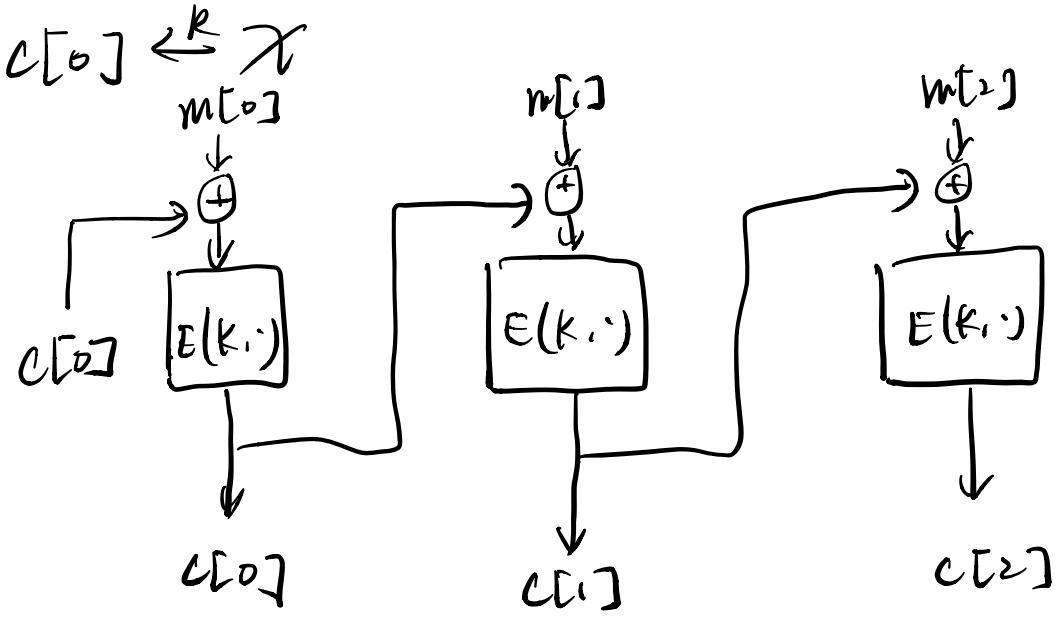
$$\rightarrow CPAadv[B, E] = |\Pr[W_0] - \Pr[W_1]|$$

$$= |P_0 - P_1 - \ell \cdot SSadv[A, E]|$$

$$\leq |P_0 - P_1| + \ell \cdot SSadv[A, E]$$

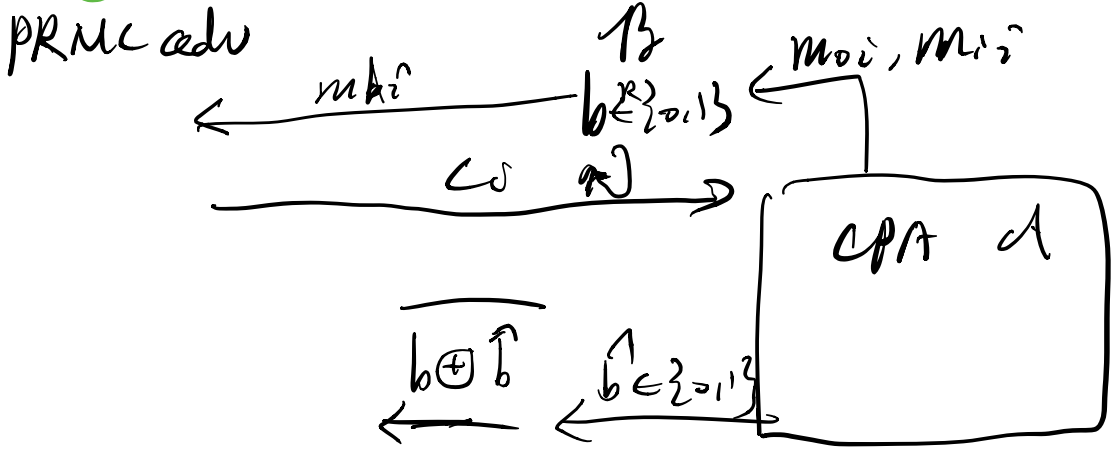
$$= PRMCadv[A, E] + \ell \cdot PRFadv[A, F]$$

(b) CBC-mode



Similar to (a).

(c) Prove PRMC → CPA



$Pr[W_0] = Pr[b = \hat{b}]$ for A.

$Pr[W_1] = \frac{1}{2}$

$$\left| Pr[W_0] - Pr[W_1] \right| = \left| Pr[b = \hat{b}] - \frac{1}{2} \right|$$

(d) Prove CPA $\not\Rightarrow$ PRMC via example

idea: sth that adversary couldn't distinguish between $E(K, m_1)$ and $E(K, m_0)$ but could distinguish $E(K, \cdot)$ & $c \xleftarrow{R} \mathcal{C}$

example:

$$E = \left(\text{RanCtr}(K, m)\Big|_{0 \le |y| \le l-1} \;\|\; \{0\}^{|y|} \right)$$

↳ i.e. truncate part of original randomized ctr mode ciphertext, then append all "0" behind.

↳ the result of this :)

5.2 (Multi-key CPA security). Generalize the definition of CPA security to the multi-key setting, analogous to Definition 5.1. In this attack game, the adversary gets to obtain encryptions of many messages under many keys. The game begins with the adversary outputting a number Q indicating the number of keys it wants to attack. The challenger chooses Q random keys. In every subsequent encryption query, the adversary submits a pair of messages and specifies under which of the Q keys it wants to encrypt; the challenger responds with an encryption of either the first or second message under the specified key (depending on whether the challenger is running Experiment 0 or 1). Flesh out all the details of this attack game, and prove, using a hybrid argument, that (single-key) CPA security implies multi-key CPA security. You should show that security degrades linearly in Q. That is, the advantage of any adversary $\mathcal{A}$ in breaking the multi-key CPA security of a scheme is at most $Q \cdot \epsilon$, where ϵ is the advantage of an adversary $\mathcal{B}$ (which is an elementary wrapper around $\mathcal{A}$) in attacking the scheme's (single-key) CPA security.

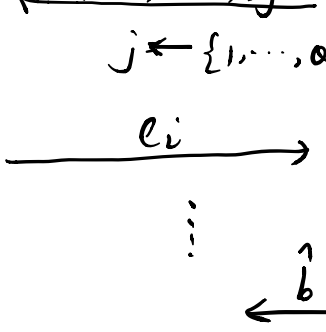
Multi-key CPA attack game:

chal.

$\{k_1, \cdots, k_Q\} \xleftarrow{R} \mathcal{K}^Q$

$b \xleftarrow{R} \{0,1\}$

$c_i \leftarrow E(k_j, m_{ib})$



adv.

$\mathrm{MCPAadv}[\mathcal{A}, \mathcal{E}]$

$= |\Pr[W_0] - \Pr[W_1]|$

where W_b is the prob. of adv. outputting 1 in $\mathrm{Exp}(b)$.

OR

$\mathrm{MCPAadv}^*[\mathcal{A}, \mathcal{E}] = \Pr(\hat{b} = b)$

(bit-guessing game)

Prove: $\mathrm{CPA} \Rightarrow \mathrm{MCPA}$ && $\mathrm{MCPAadv}[\mathcal{A}, \mathcal{E}] \leq Q \cdot \mathrm{CPAadv}[\mathcal{B}, \mathcal{E}]$

Proof Idea: using hybrid argument, imagine in "Game 1", all keys are the same (i.e. all j shares the same value), then "Game 0" is effectively the original CPA game; whereas in "Game Q", it'll be MCPA game;

between any "Game k" and "Game (k+1)", the differences only embeds in

$\{k_1, k_2, \cdots k_k\}$ $\{k_1, k_2, \cdots k_k, k_{k+1}\}$

cases where $j = k+1$,

Noted that all $k_1, k_2 \cdots k_k$ (k_{k+1}) are independently, randomly chosen from $\mathcal{K}$ space. thus the statistical distance of "c_i" between 2 adjacent Games is solely determined by $E(k, \cdot)$'s output distribution over choice of $k_{k+1} \xleftarrow{R} \mathcal{K}$, which is exactly the statistical distance between $\mathrm{Exp}(0)$ and $\mathrm{Exp}(1)$ of a CPA game!!

$\Rightarrow |P_{k+1} - P_k| \leq \Delta[P_{(\mathrm{CPA}, b=0)} - P_{(\mathrm{CPA}, b=1)}] \leq \mathrm{CPAadv}[\mathcal{B}, \mathcal{E}]$

$\Rightarrow \mathrm{MCPAadv}[\mathcal{A}, \mathcal{E}] = P_Q \leq |P_Q - P_{Q-1}| + |P_{Q-1} - P_{Q-2}| + \cdots + |P_2 - P_1| + P_1 \leq Q \cdot \mathrm{CPAadv}[\mathcal{B}, \mathcal{E}]$

5.5 (A simple proof of randomized counter mode security). As mentioned in Remark 5.3, we can view randomized counter mode as a special case of the generic hybrid construction in Section 5.4.1. To this end, let F be a PRF defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, where $\mathcal{X} = \{0, \ldots, N-1\}$ and $\mathcal{Y} = \{0,1\}^n$, where N is super-poly. For poly-bounded $\ell \geq 1$, consider the PRF F' defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y}^\ell)$ as follows:

$$F'(k,x) := \Big(F(k,x),\ F(k, x+1 \bmod N),\ \ldots,\ F(k, x+\ell-1 \bmod N) \Big).$$

(a) Show that F' is a weakly secure PRF, as in Definition 4.3.

(b) Using part (a) and Remark 5.2, give a short proof that randomized counter mode is CPA secure.

(a)

Weakly secure PRFs. For certain constructions that use PRFs it suffices that the PRF satisfy a weaker security property than Definition 4.2. We say that a PRF is *weakly secure* if no efficient adversary can distinguish the PRF from a random function when its queries are severely restricted: it can only query the function at *random* points in the domain. Restricting the adversary's queries to random inputs makes it potentially easier to build weakly secure PRFs. In Exercise 4.2 we examine natural PRF constructions that are weakly secure, but not fully secure.

We define weakly secure PRFs by slightly modifying Attack Game 4.2. Let F be a PRF defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$. We modify the way in which an adversary $\mathcal{A}$ interacts with the challenger: whenever the adversary queries the function, the challenger chooses a random $x \in \mathcal{X}$ and sends both x and $f(x)$ to the adversary. In other words, the adversary sees evaluations of the function f at *random* points in $\mathcal{X}$ and needs to decide whether the function is truly random or pseudorandom. We define the adversary's advantage in this game, denoted $\mathsf{wPRFadv}[\mathcal{A}, F]$, as in (4.21).

Proof Idea: first question, why $F'(k,x)$ is not fully secure PRF?

$y_1 \leftarrow F'(k,x)$ 2 queries $\Rightarrow$ $y_1[1] = y_2[0]$

$y_2 \leftarrow F'(k,x+1)$ $y_1[2] = y_2[1]$

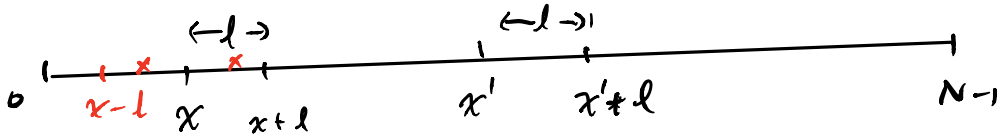
$\vdots$

$y_1[i] = y_2[i-1]$ $\quad i = 1, \ldots, l-1$

Secondly, why $F'(k,x)$ is weakly secure then?

if x is randomly chosen, then the likelihood of overlapping is small. range

i.e.



further, what's the prob. of overlap?

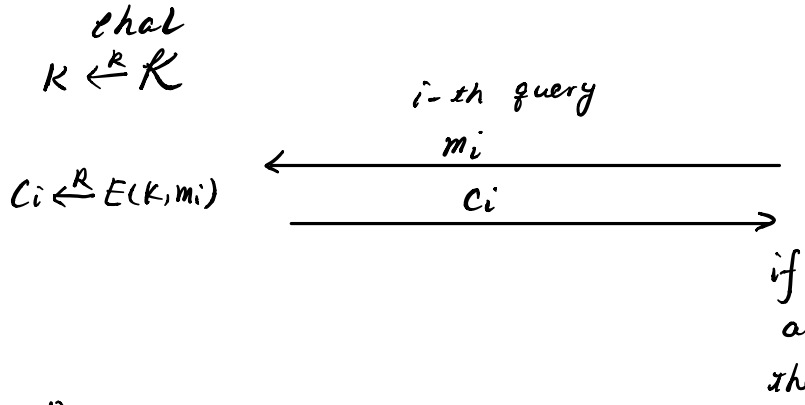
↳ every new x would effectively blocks x of $2l$ (see red).

(a) $\mathsf{PRFadv}[\mathcal{A}, F'] \leq l \cdot \mathsf{PRFadv}[\mathcal{B}, F] + \frac{2lQ}{N}$

(b) Since the randomized ctr-mode only evaluates IV chosen at random, the generic construction would still be secure even w/ a weakly secure PRF.

(the conclusion is very natural and proof is quite trivial, thus details omitted)

5.11 (Repeating ciphertexts). Let $\mathcal{E} = (E, D)$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Assume that there are at least two messages in $\mathcal{M}$, that all messages have the same length, and that we can efficiently generate messages in $\mathcal{M}$ uniformly at random. Show that if $\mathcal{E}$ is CPA secure, then it is infeasible for an adversary to make an encryptor generate the same ciphertext twice. The precise attack game is as follows. The challenger chooses $k \in \mathcal{K}$ at random and the adversary makes a series of queries; the ith query is a message m_i, to which the challenger responds with $c_i \xleftarrow{R} E(k, m_i)$. The adversary wins the game if any two c_i's are the same. Show that if $\mathcal{E}$ is CPA secure, then every efficient adversary wins this game with negligible probability. In particular, show that the advantage of any adversary $\mathcal{A}$ in winning the repeated-ciphertext attack game is at most 2ϵ, where ϵ is the advantage of an adversary $\mathcal{B}$ (which is an elementary wrapper around $\mathcal{A}$) that breaks the scheme's CPA security.



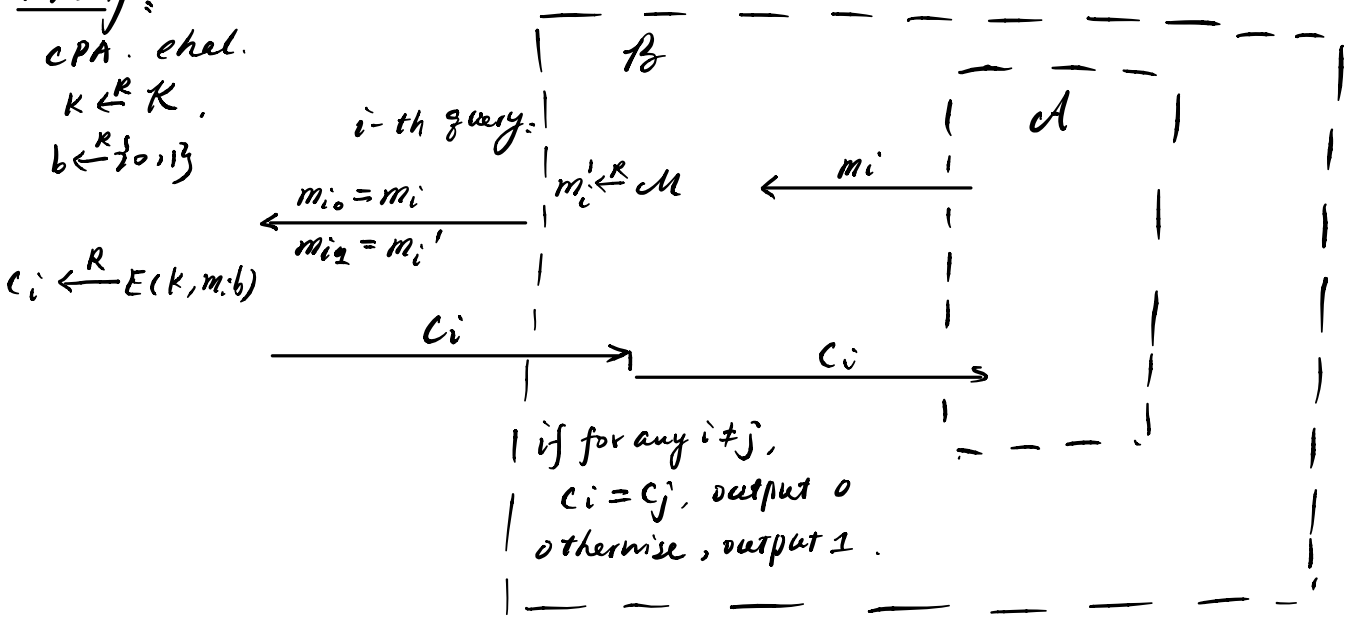
Prove:

$RCadv[A, \mathcal{E}] \leq 2 \cdot CPAadv[B, \mathcal{E}]$

i.e. CPA $\Rightarrow$ RC secure.

Proof Idea: construct an elementary wrapper to show: if such RC adv exists, then an efficient $\mathcal{B}$ could succeed in attacking CPA game. intuitively, if repeated-ciphertext is found, then in CPA game is distinguishable.

Proof:



$Pr[W_0] = Pr[A \text{ wins}]$

$Pr[W_1] = \frac{1}{2}$

$CPA^{\#}adv[\mathcal{B}, \mathcal{E}] = |Pr[W_0] - Pr[W_1]| = |Pr[A \text{ wins}] - \frac{1}{2}| = RCadv[A, \mathcal{E}]$

✓

5.13 (CBC encryption with small blocks is insecure). Suppose the block cipher used for CBC encryption has a block size of n bits. Construct an attacker that wins the CPA game against CBC that makes $\approx 2^{n/2}$ queries to its challenger and gains an advantage $\approx 1/2$. Your answer explains why CBC cannot be used with a block cipher that has a small block size (e.g. $n = 64$ bits). This is one reason why AES has a block size of 128 bits.

better way ?

Discussion: This attack was used to show that 3DES is no longer secure for Internet use, due to its 64-bit block size [17].

Idea: this is a direct application of "the birthday paradox". !!

just by looking at Eq. 5.22:

$$\mathrm{CPAadv}[\mathcal{A},\mathcal{E}'] \le \frac{2\sigma^2 l^2}{N} + 2\cdot \mathrm{BCadv}[\mathcal{B},\mathcal{E}]$$

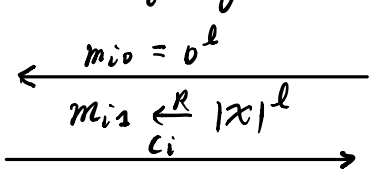
$$= \frac{2\cdot (2^{n/2})^2 \cdot l^2}{2^n} + 2\cdot \mathrm{BCadv}[\mathcal{B},\mathcal{E}]$$

$$= 2l^2 + 2\cdot \mathrm{BCadv}[\mathcal{B},\mathcal{E}]$$

↑ obviously non-negligible.

Next, we need to construct an adversary that exploits this fact — "how do we transform this collision finding ability into distinguishability semantically?"

Proof:



Let $l = 1$ for simplicity.

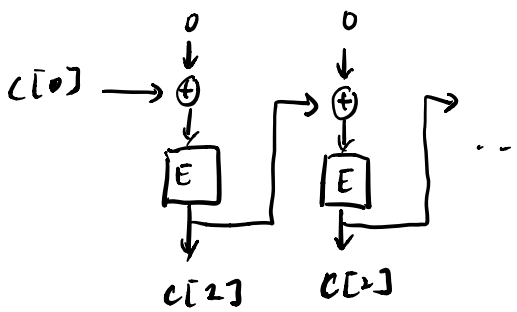
for $c_i[0] = c_j[0]$, remove {either c_i or c_j} $(i \neq j)$

after removal, if $\exists\ c_i = c_j$ for $(i \neq j)$

output 1,

otherwise, output 0.

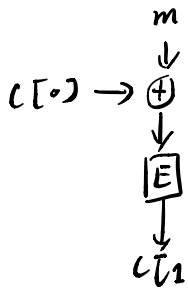
in Exp(0),



After removing c_i, all input to BC is unique, thus:

$$\Pr[c_i = c_j,\ i \neq j] \approx \Pr[E(K,x_1) = E(K,x_2) \mid x_1 \neq x_2] \approx \mathrm{BCadv}[\mathcal{A},F]$$

in Exp(1):



Note: removing c_i s.t. $c_i[0] = c_j[0]$ does little effect on birthday collision.

w.l.o.g. $\approx \sqrt{|x|}$.

$$\Pr[\exists i,j, \text{s.t. } c_i \oplus m_i = c_j \oplus m_j \text{ and } c_i[0] \neq c_j[0]] \approx 50\%$$

$$\Rightarrow |\Pr[W_0] - \Pr[W_1]| \approx 50\%$$

5.14 (An insecure nonce-based CBC mode). Consider the nonce-based CBC scheme $\mathcal{E}'$ described in Section 5.5.3. Suppose that the nonce space $\mathcal{N}$ is equal to block space $\mathcal{X}$ of the underlying block cipher $\mathcal{E} = (E, D)$, and the PRF F is just the encryption algorithm E. If the two keys k and k' in the construction are chosen independently, the scheme is secure. Your task is to show that if only one key k is chosen, and the other key k' is set to k, then the scheme is insecure.

here's the nonce-based CBC attack game:

chal. adv

$b \xleftarrow{R} \{0,1\}$

$K \xleftarrow{R} \mathcal{K}$

$K' \xleftarrow{R} \mathcal{K}'$

i-th query

$\xleftarrow{m_{i0}, m_{i1}, N_i}$

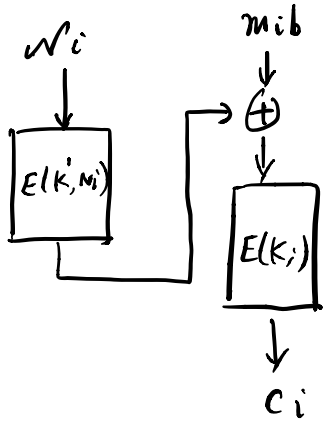
$IV_i \leftarrow E(K', N_i)$

$C_i \leftarrow BC(K, m_{ib}, IV_i)$

where IV is initial vector

$\xrightarrow{C_i}$

$\xleftarrow{\hat{b} \in \{0,1\}}$



Proof Idea:

if $K = K'$, then

$E(K, N_i) \oplus m_i \longrightarrow C_i$ known

next time

$(E(K, N_i) \oplus m_i) \oplus 0 \longrightarrow C_i$ ✓

predictable ✓

Proof:

(m_{10}, m_{11}, N_1)

(m_{20}, m_{21}, N_2)

let $m_{10}, m_{11} \xleftarrow{R} \mathcal{M}$

$N_1 \xleftarrow{R} \mathcal{N}$

upon getting C_1,

let $m_{20} = C_1$, $m_{21} \xleftarrow{R} \mathcal{M}$,

$N_2 \leftarrow 0$

$\Rightarrow$ if $Exp(0)$, then $C_1 = C_2$

if $Exp(1)$, then $C_1 \neq C_2$.

$\Rightarrow$ perfect distinguishability

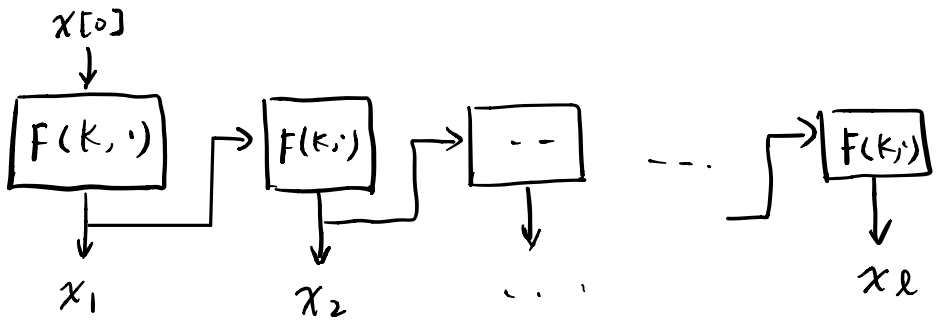
n CPA adv $= 1$.

5.15 (Output feedback mode). Suppose F is a PRF defined over $(\mathcal{K}, \mathcal{X})$, and $\ell \geq 1$ is poly-bounded.

(a) Consider the following PRG $G : \mathcal{K} \to \mathcal{X}^\ell$. Let x_0 be an arbitrary, fixed element of $\mathcal{X}$. For $k \in \mathcal{K}$, let $G(k) := (x_1, \ldots, x_\ell)$, where $x_i := F(k, x_{i-1})$ for $i = 1, \ldots, \ell$. Show that G is a secure PRG, assuming F is a secure PRF and that $|\mathcal{X}|$ is super-poly.

(b) Next, assume that $\mathcal{X} = \{0,1\}^n$. We define a cipher $\mathcal{E} = (E, D)$, defined over $(\mathcal{K}, \mathcal{X}^\ell, \mathcal{X}^{\ell+1})$, as follows. Given a key $k \in \mathcal{K}$ and a message $(m_1, \ldots, m_\ell) \in \mathcal{X}^\ell$, the encryption algorithm E generates the ciphertext $(c_0, c_1, \ldots, c_\ell) \in \mathcal{X}^{\ell+1}$ as follows: it chooses $x_0 \in \mathcal{X}$ at random, and sets $c_0 = x_0$; it then computes $x_i = F(k, x_{i-1})$ and $c_i = m_i \oplus x_i$ for $i = 1, \ldots, \ell$. Describe the corresponding decryption algorithm D, and show that $\mathcal{E}$ is CPA secure, assuming F is a secure PRF and that $|\mathcal{X}|$ is super-poly.

Note: This construction is called **output feedback mode** (or **OFB**).



$K \xleftarrow{R} \mathcal{K}$.
$x[0]$ fixed.

(a) Proof Idea: for every block, if PRF is secure, x_i should be indist. from random $x \in \mathcal{X}$. When putting all chained blocks together, this PRG might reveal a tail when $x_i = x_j \Rightarrow x_{i+1} = x_{j+1}$, but such occurance is rare, thus PRG should be secure.

Proof: using Hybrid arguments:

W_b: outputting 1 in PRG game b.

Game 1: just the original game. $P_0 = \Pr[W_0]$

Game l: replacing all $F(K,\cdot)$ with truly random generator. $P_l = \Pr[W_1]$

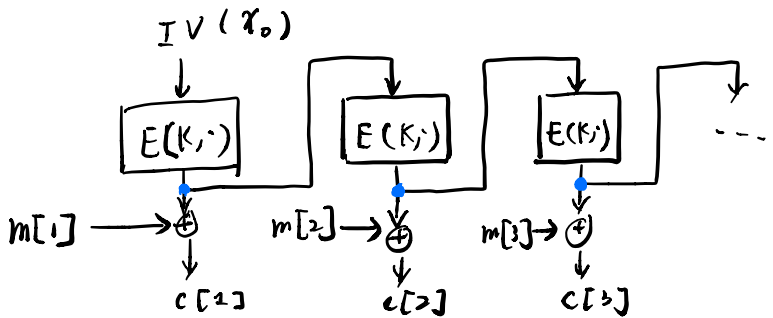
$|P_j - P_{j-1}| = \text{PRFadv}[A, F]$

PRG dist. game in Game 1 : $|P_0 - \frac{1}{2}| = \Pr[x_i = x_j, x \neq j \text{ for truly random}]$

$\leq \frac{q^2 l^2}{|\mathcal{X}|}$

$\rightarrow \text{PRGadv}[A, G] \leq (l-1)\text{PRFadv}[A, F] + \frac{q^2 l^2}{|\mathcal{X}|}$

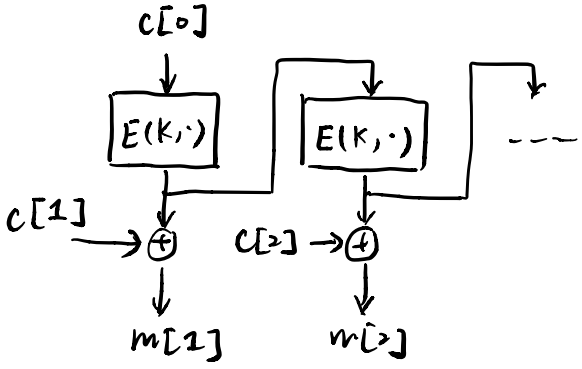
(b)



(output feedback mode) OFB

$C = x_0 c_1 c_2 \cdots c_l$.

here's decryption:



Proof: OFB is CPA secure.

Proof Idea:

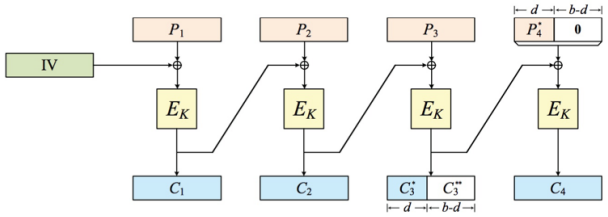
based on the secure PRG we prove in (a), we knew that output in OFB at blue points are indist. from any random bitstring of length $|x|$. thus by further XOR with the message block, we would end up with a CPA secure cipher.

Further, since IV is randomized, the $\mathcal{E}$ cipher is also probabilistic.

NOTE: only Encryption in $\mathcal{E} = (E, D)$ is used, both for encryption and decryption.

Proof: (since proof idea almost explain all, detail proofs should be trivial from there).

5.16 (CBC ciphertext stealing). One problem with CBC encryption is that messages need to be padded to a multiple of the block length and sometimes a dummy block needs to be added. The following figure describes a variant of CBC that eliminates the need to pad:

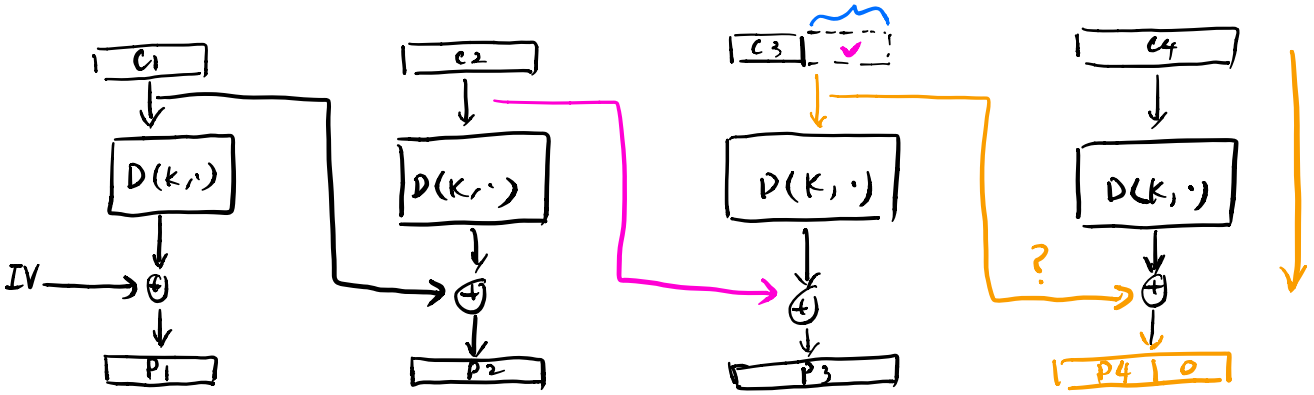


The method pads the last block with zeros if needed (a dummy block is never added), but the output ciphertext contains only the shaded parts of C_1, C_2, C_3, C_4. Note that, ignoring the IV, the ciphertext is the same length as the plaintext. This technique is called *ciphertext stealing*.

(a) Explain how decryption works.

(b) Can this method be used if the plaintext contains only one block? b-d

(a)



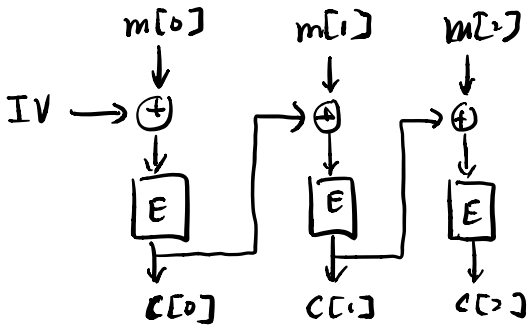
step1 : calculate/observe length of padding through truncated ciphertext. (b-d)

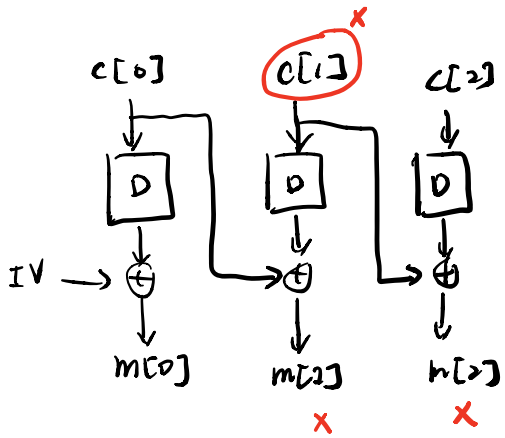
step2 : decrypt C_4 (last ciphertext block), and find out the (b-d) LSB of original C_3.

step3 : fill in the missing C_3 and the rest is just the same as a normal CBC decryption.

(b) NO, since there's no "second last block" to steal from.

5.17 (Single ciphertext block corruption in CBC mode). Let c be an ℓ block CBC-encrypted ciphertext, for some $\ell > 3$. Suppose that exactly one block of c is corrupted, and the result is decrypted using the CBC decryption algorithm. How many blocks of the decrypted plaintext are corrupted?

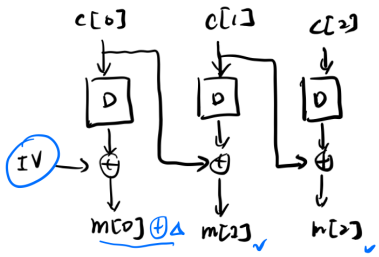




shown above, since CBC is chained, any corruption at $c[i]$ $(i = 0, \cdots, \ell-1)$, would caused all msg blocks to be incorrectly decrypted. thus $(\ell - i)$ would be affected. assuming index starts at 0 by convention.

NOTE: I assume in practice, mitigation / best practice would use authenticated encryption with integrity check, so that any corrupted ciphertext would fail the integrity check and won't be decrypted at all.

5.18 (The malleability of CBC mode). Let c be the CBC encryption of some message $m \in \mathcal{X}^\ell$, where $\mathcal{X} := \{0,1\}^n$. You do not know m. Let $\Delta \in \mathcal{X}$. Show how to modify the ciphertext c to obtain a new ciphertext c' that decrypts to m', where $m'[0] = m[0] \oplus \Delta$, and $m'[i] = m[i]$ for $i = 1, \ldots, \ell - 1$. That is, by modifying c appropriately, you can flip bits of your choice in the first block of the decryption of c, without affecting any of the other blocks.



change IV however you want
let $IV' = IV \oplus \Delta$.
then requirement met.

→ more generally, if we want to modify content in block N, and don't care about unpredicable decrypted msg at $N-1$, then change ciphertext at $N-1$.

NOTE: once again, lessons learned here is use "authenticated encryption" in practice !!

5.19 (Online ciphers). In practice there is a strong desire to encrypt one block of plaintext at a time, outputting the corresponding block of ciphertext right away. This lets the system transmit ciphertext blocks as soon as they are ready without having to wait until the entire message is processed by the encryption algorithm. (streaming block cipher)

(a) Define a CPA-like security game that captures this method of encryption. Instead of forcing the adversary to submit a complete pair of messages in every encryption query, the adversary should be allowed to issue a query indicating the beginning of a message, then repeatedly issue more queries containing message blocks, and finally issue a query indicating the end of a message. Responses to these queries will include all ciphertext blocks that can be computed given the information given.

(b) Show that randomized CBC encryption is not CPA secure in this model.

(c) Show that randomized counter mode is online CPA secure.

$m = |x|^\ell$

(a)

chal. adv.

$k \xleftarrow{R} \mathcal{K}$.

$b \xleftarrow{R} \{0,1\}$

$\xrightarrow{\quad}$ $c[0]$

$\xleftarrow{\quad}$ $m_0, m_1 \in \mathcal{X}$, $i = (1, 2, \cdots, \ell-1)$

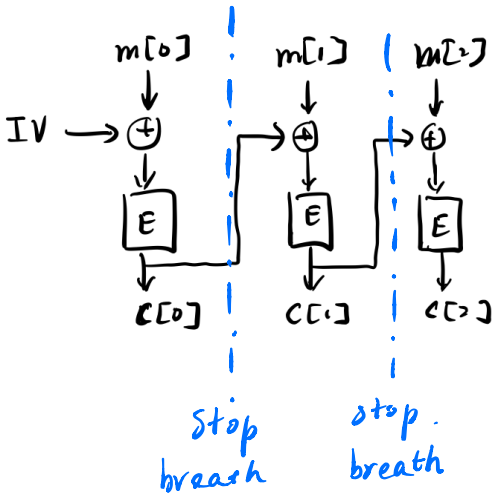
$c[i] \xleftarrow{R} E(K, m_b[i])$

cache $c[i-1]$ or any previous intermediary output if necessary.

$\xrightarrow{\quad}$ $c[i]$

$\xleftarrow{\quad}$ $\hat{b} \in \{0,1\}$.

(b) Proof Idea: what changed? → adversary is more adaptive as they have visibility of all intermediary before sending their next query.



let $m_0[0] \xleftarrow{R} \mathcal{X}$ (E over $\mathcal{K} * \mathcal{X}$.)

$m_1[0] \xleftarrow{R} \mathcal{X}$

$m_0[0] \neq m_1[0]$

upon receiving IV and $c[0]$.

let $m_0[1] = IV \oplus c[0] \oplus m_0[0]$

$m_1[1] \xleftarrow{R} \mathcal{X}$

then, if we're in Exp(0).

$c[1]$ will be equal to $c[0]$

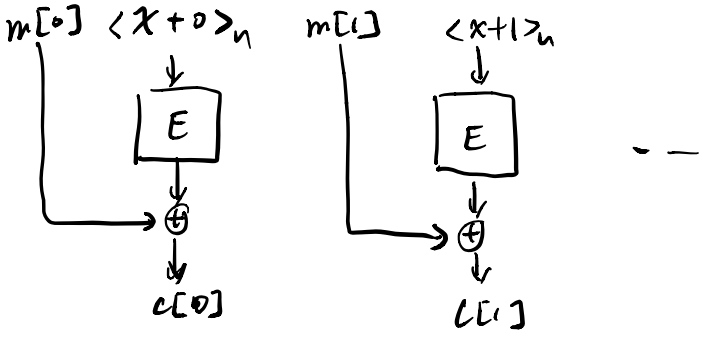
else in Exp(1):

$c[1]$ will be some unpredicable hex string.

$\Rightarrow$ perfect distri.

NOT CPA secure

(c)



Proof Idea:

What's different about ctr-mode to be secure in contrast to poor CBC construction?

↳ answer: every block is independent of each other (thus its computation parallelizable).

they all only depend on X, which is unknown to attacker, and further all ctr went through PRF before XOR w/ msg block. which leaves no extra benefit / advantage for being adaptive.

Proof: every new cipher block is a typical analysis with PRF security.

no useful info can be learned or leveraged.

(about previous cipher block)

Details emitted.

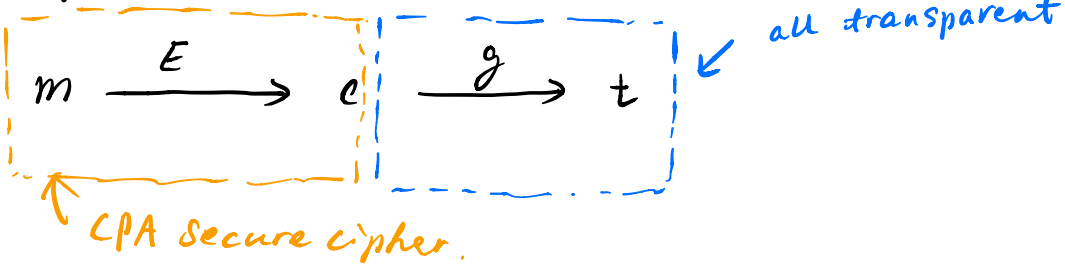
5.20 (Redundant bits do not harm CPA security). Let $\mathcal{E} = (E, D)$ be a CPA-secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Show that appending to a ciphertext additional data that is computed from the ciphertext does not damage CPA security. Specifically, let $g : \mathcal{C} \to \mathcal{Y}$ be some efficiently computable function. Show that the following modified cipher $\mathcal{E}' = (E', D')$ is CPA-secure:

$$E'(k, m) := \{c \leftarrow E(k, m),\ t \leftarrow g(c),\ \text{output } (c, t)\}$$
$$D'\big(k,\ (c, t)\big) := D(k, c)$$

NOTE: on first sight, this conclusion could be very helpful for "adding integrity tag for ciphertext", which will be explored in Chapter 9.

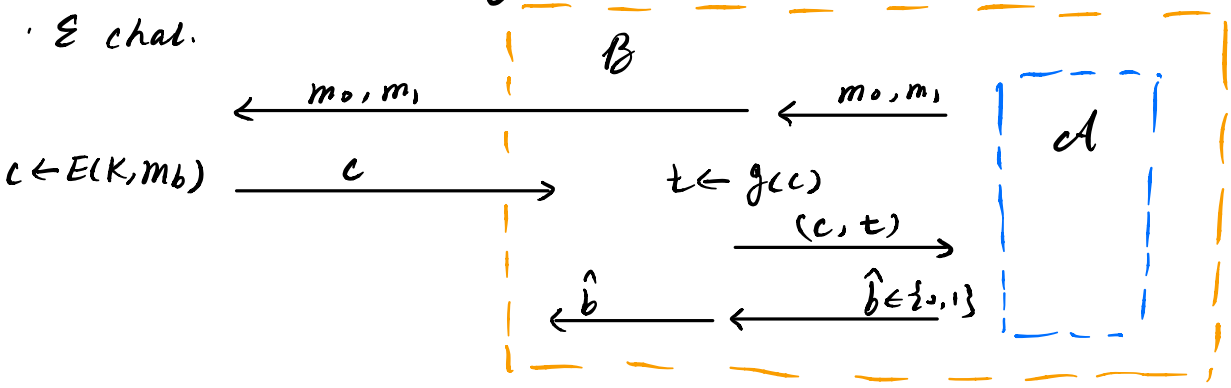
Proof Idea:

since g is a pubically known function, it doesn't give extra info about (m, c) pair, in another word:



Proof: using elementary wrapper, trying to prove: $\mathcal{E} \longrightarrow \mathcal{E}'$ CPA

$\mathcal{E}$ chal.



it's obvious that

$$\text{CPAadv}^*[\mathcal{B}, \mathcal{E}] = \text{CPAadv}^*[\mathcal{A}, \mathcal{E}']$$

5.21 (Broadcast encryption). In a broadcast encryption system, a sender can encrypt a message so that only a specified set of recipients can decrypt. Such a system is made up of three efficient algorithms (G, E, D): algorithm G is invoked as $G(n)$ and outputs an encryptor key ek, and n keys $k_1, \ldots, k_n$, one key for each recipient; algorithm E is invoked as $c \xleftarrow{R} E(ek, m, S)$, where m is the message and $S \subseteq \{1, \ldots, n\}$ is the intended set of recipients; algorithm D is invoked as $m \leftarrow D(k_i, c)$ for some $1 \le i \le n$, and correctly decrypts the given c whenever i is in the set S. More precisely, for all m and all subsets S of $\{1, \ldots, n\}$, we have that $D\big(k_i,\ E(ek, m, S)\big) = m$ for all $i \in S$.

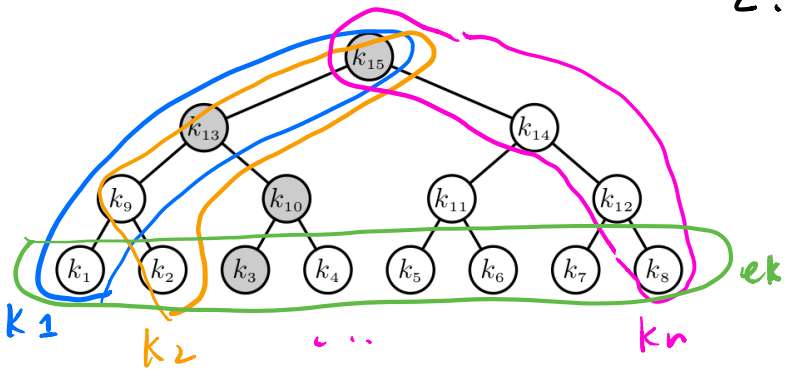
(a) Describe the revocation scheme described in (5.35) in Section 5.6 as a broadcast encryption system. How do algorithms G, E, D work and what are ek and $k_1, \ldots, k_n$?

(b) A broadcast encryption scheme is secure if a set of colluding recipients B learns nothing about plaintexts encrypted for subsets of $\{1, \ldots, n\} \setminus B$, namely plaintexts that are not intended for the members of B. More precisely, CPA security of a broadcast encryption system is defined using the following two experiments, Experiment 0 and Experiment 1: In Experiment b, for $b = 0, 1$, the adversary begins by outputing a subset B of $\{1, \ldots, n\}$. The challenger then runs $G(n)$ and sends to the adversary all the keys named in B, namely $\{k_i\}_{i \in B}$. Now the adversary issues chosen plaintext queries, where query number j is a triple $(S_j, m_{j,0}, m_{j,1})$ for some set S_j in $\{1, \ldots, n\} \setminus B$. The challenger sends back $c_j \xleftarrow{R} E(ek, m_{j,b}, S_j)$. The system is secure if the adversary cannot distinguish these two experiments.

Show that the scheme from part (a) is a secure broadcast encryption system, assuming the underlying header encryption scheme is CPA secure, and the body encryption scheme (E', D') is semantically secure.

Hint: Use a sequence of $2n - 1$ hybrids, one for each key in the tree of Fig. 5.5

(a) $G(n)$:



E:

$$c_m := \left\{ \begin{array}{l} k \xleftarrow{R} \mathcal{K} \\ \text{for } u \in \text{cover}(S) : c_u \xleftarrow{R} E(k_u,\ k) \\ c \xleftarrow{R} E'(k, m) \\ \text{output } (\{c_u\}_{u \in \text{cover}(S)},\ c) \end{array} \right\}.$$

$D(k_i, c_m) = m$

↳ have to choose the right key in k_i to decrypt k to further get the m.

(b) Proof Idea:

upon every revocation, keys along the path from the corrupted leaf to the root will be nulled, and future msg will be encrypted using their siblings.

? (quite obvious, don't know how to formalize)