Observe that a block cipher can be used to construct a PRG:

$F: \{0,1\}^\lambda \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a block cipher

Define $G: \{0,1\}^\lambda \longrightarrow \{0,1\}^{\ell n}$ as

$$G(k) = \underbrace{F(k,1) \,\|\, F(k,2) \,\|}_{\uparrow \text{ string concatenation}} \cdots \| \, F(k,\underbrace{\ell}_{\uparrow \text{ write input as an } n\text{-bit string}})$$

← this stream cipher allows random access!

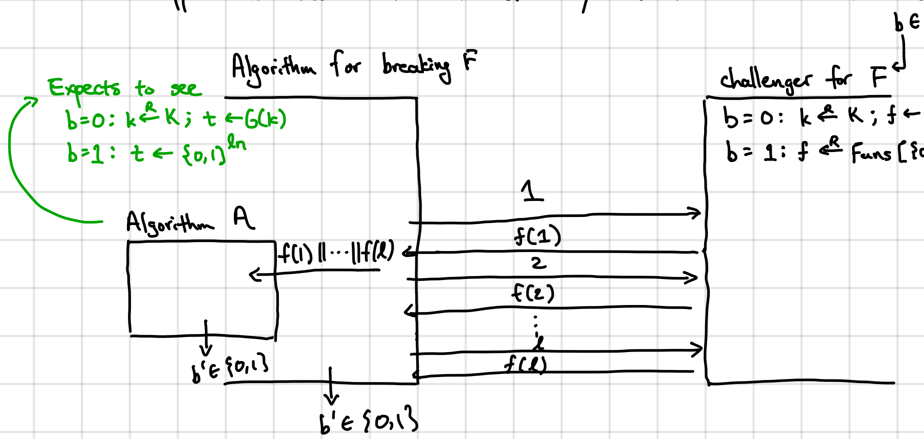
(just require that $n > \log \ell$)

we said PRP above (will revisit this)

Theorem. If F is a secure PRF, then G is a secure PRG.

Proof. As usual, we show the contrapositive: if G is not a secure PRG, then F is not a secure PRF.

Suppose we have efficient adversary A for G. We use A to build adversary for F:



1. If $\ell = \text{poly}$, then B is efficient
2. If $b=0$: B sends $G(k)$ to A where k is a uniformly random key

   If $b=1$: B sends uniformly random string (f is random function) to A
3. $\text{PRFAdv}[B,F] = \left| \Pr[b'=1 \mid b=0] - \Pr[b'=1 \mid b=1] \right|$

$= \left| \Pr[A \text{ outputs } 1 \mid b=0] - \Pr[A \text{ outputs } 1 \mid b=1] \right|$

$= \text{PRGAdv}[A,G]$

which is non-negligible by assumption.

But... we used a block cipher (PRP) in our construction above. Does the proof still go through?

Not quite...

for a random function, $f(1) = f(2)$ with probability $\frac{1}{2^n}$

for a random permutation, $f(1) = f(2)$ with probability 0

but 2^{-n} might be very very small... adversary won't notice unless it sees a "collision" [i.e., two values x, y where $f(x) = f(y)$]

PRF Switching Lemma. Let $F: K \times X \to X$ be a secure PRP. Then, for any Q-query adversary A:

$$\left| \text{PRPAdv}[A,F] - \text{PRFAdv}[A,F] \right| \leq \frac{Q^2}{2|X|}$$

Proof Idea. Adversary essentially cannot tell the difference unless it sees a collision. If there is no collision, then it is just seeing random values. How many queries before there is a collision? Birthday paradox: $Q \sim \sqrt{|X|}$

Take-away: If $|X|$ is large (e.g., exponential), then we can use a PRP as a PRF.

- 3DES: $n = 64$ so $|X| = 2^{64}$ [if adversary makes $\ll 2^{32}$ queries, then can use it as a PRF]
- AES: $n = 128$ so $|X| = 2^{128}$ [if adversary makes $\ll 2^{64}$ queries, then can use it as a PRF]

<u>Thus far</u>: PRP/PRF in "counter mode" gives us a stream cipher (one-time encryption scheme)

<u>How do we reuse it?</u> Choose a <u>random</u> starting point (called an initialization vector) ← typically, the IV is divided into a nonce (value that does not repeat) and a counter : IV = nonce ‖ counter

"randomized counter mode"

		m_1	m_2	m_3	m_4
$\oplus$	IV (random value)	$F(k, IV)$	$F(k, IV+1)$	$F(k, IV+2)$	$F(k, IV+3)$

divide message into blocks (based on block size of PRF)

IV	c_1	c_2	c_3	c_4

ciphertext

<u>observe</u>: ciphertext is longer than the message (required for CPA security)

<u>Theorem</u>: Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a secure PRF and let Π_{CTR} denote the randomized counter mode encryption scheme from above for ℓ-block messages $(\mathcal{M} = \mathcal{X}^{\leq \ell})$. Then, for all efficient CPA adversaries $\mathcal{A}$, there exists an efficient PRF adversary $\mathcal{B}$ such that

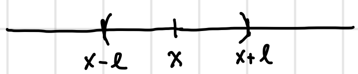
$$\text{CPAAdv}[\mathcal{A}, \Pi_{CTR}] \leq \frac{4Q^2\ell}{|\mathcal{X}|} + 2 \cdot \text{PRFAdv}[\mathcal{B}, F]$$

Q: number of encryption queries

ℓ: number of blocks in message

<u>Intuition</u>:

1. If there are no collisions (i.e., PRF never evaluated on the same block), then it is as if everything is encrypted under a fresh one-time pad.
2. Collision event: $(x, x+1, \ldots, x+\ell-1)$ overlaps with $(x', x'+1, \ldots, x'+\ell-1)$ when $x, x' \xleftarrow{R} \mathcal{X}$



probability that x' lies in this interval is $\leq \frac{2\ell}{|\mathcal{X}|}$

There are $\leq Q^2$ possible pairs (x, x'), so by a union bound,

$$\Pr[\text{collision}] \leq \frac{2\ell Q^2}{|\mathcal{X}|}$$

3. Remaining factor of 2 in advantage due to intermediate distribution (hybrid argument):

Encrypt m_0 with PRF

$\quad\quad \text{PRFAdv}[\mathcal{B}, F] + \frac{2\ell Q^2}{|\mathcal{X}|}$

Encrypt m_0 with fresh one-time pad

$\quad\quad 0$

Encrypt m_1 with fresh one-time pad

$\quad\quad \text{PRFAdv}[\mathcal{B}, F] + \frac{2\ell Q^2}{|\mathcal{X}|}$

Encrypt m_1 with PRF

<u>Interpretation</u>: If $|\mathcal{X}| = 2^{128}$ (e.g., AES), and messages are 1 MB long (2^{16} blocks) and we want the distinguishing advantage to be below 2^{-32}, then we can use the same key to encrypt

$$Q \leq \sqrt{\frac{|\mathcal{X}| \cdot 2^{-32}}{4\ell}} = \sqrt{\frac{2^{96}}{2^{18}}} = \sqrt{2^{78}} = 2^{39}$$

(~ 1 trillion messages!)

Nonce-based counter mode: divide IV into two pieces: IV = nonce ‖ counter

↑ value that does not repeat

common choices: 64-bit nonce, 64-bit counter; 96-bit nonce, 32-bit counter } only nonce needs to be sent! (slightly smaller ciphertexts)

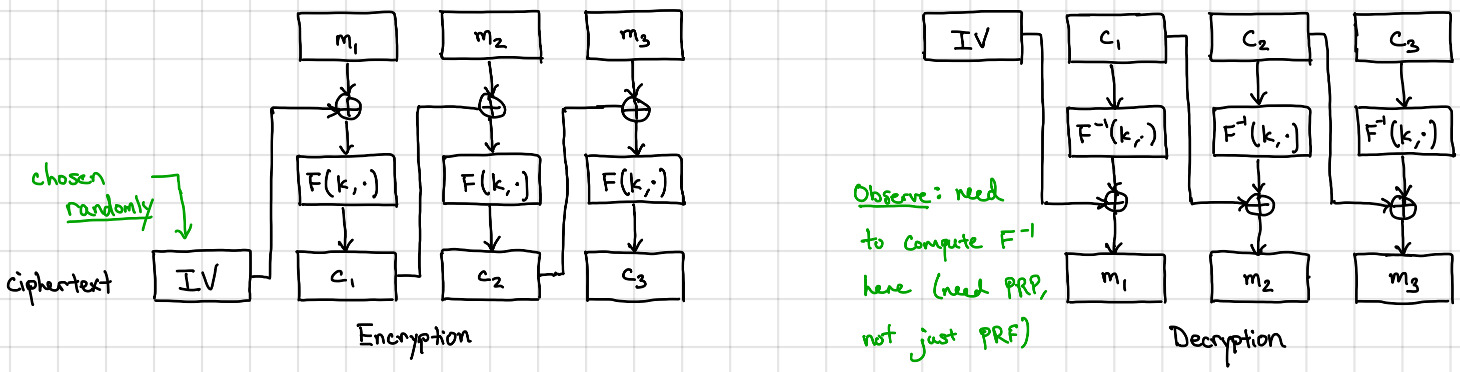
Only requirement for security is that IV does not repeat:

- Option 1: Choose randomly (either IV or nonce)
- Option 2: If sender + recipient have shared state (e.g., packet counter), can just use a counter, in which case, IV/nonce does not have to be sent

(CTR)
Counter mode is parallelizable, simple-to-implement, just requires PRF — preferred mode of using block ciphers

Other block cipher modes of operation:

Cipherblock chaining (CBC): common mode in the past (e.g., TLS 1.0, still widely used today)



Theorem: Let $F: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a secure PRF and let Π_{CBC} denote the CBC encryption scheme for ℓ-block messages ($\mathcal{M} = \mathcal{X}^{\leq \ell}$). Then, for all efficient CPA adversaries $\mathcal{A}$, there exists an efficient PRF adversary $\mathcal{B}$ such that

$$\mathrm{CPAAdv}[\mathcal{A}, \Pi_{CBC}] \leq \frac{2Q^2\ell^2}{|\mathcal{X}|} + 2 \cdot \mathrm{PRFAdv}[\mathcal{B}, F]$$

Q: number of encryption queries

ℓ: number of blocks in message

Intuition: similar to analysis of randomized counter mode:

1. Ciphertext is indistinguishable from random string if PRP is evaluated on distinct inputs
2. When encrypting, PRP is invoked on ℓ random blocks, so after Q queries, we have $Q\ell$ random blocks.
   $\Rightarrow$ Collision probability $\leq \frac{Q^2\ell^2}{|\mathcal{X}|}$ ← this is larger than collision prob. for randomized counter mode by a factor of $\frac{\ell}{2}$ [overlap of Q random intervals vs. $Q\ell$ random points]
3. Factor of 2 arises for same reason as before

Interpretation. CBC mode provides weaker security compared to counter mode: $\frac{2Q^2\ell^2}{|\mathcal{X}|}$ vs. $\frac{4Q^2\ell}{|\mathcal{X}|}$

Concretely: for same parameters as before (1 MB messages, 2^{-32} distinguishing advantage):

$$Q \leq \sqrt{\frac{|\mathcal{X}| \cdot 2^{-32}}{2\ell^2}} = \sqrt{\frac{2^{128} \cdot 2^{-32}}{2(2^{16})^2}} = \sqrt{2^{63}} = 2^{31.5} \quad (\sim 1 \text{ billion messages})$$

↳ $2^{7.5}$ ~ 180x smaller than using counter mode

Padding in CBC mode: each ciphertext block is computed by feeding a message block into the PRP

$\Rightarrow$ message must be an even multiple of the block size

$\Rightarrow$ when used in practice, need to pad messages

Can we pad with zeroes? Cannot decrypt! What if original message ended with a bunch of zeroes?

Requirement: padding must be invertible

CBC padding in TLS 1.0: if k bytes of padding is needed, then append k bytes to the end, with each byte set to $k-1$

(for AES-CBC) if 0 bytes of padding is needed, then append a block of 16 bytes, with each byte equal to 15

↳ dummy block needed to ensure pad is invertible

[injective functions must expand: $|\{0,1\}^{\leq 256}| > |\{0,1\}^{256}|$]

↳ called PKCS#5/PKCS#7 (public-key cryptography standards)

Need to pad in CBC encryption can be exploited in "padding oracle" attacks

Padding in CBC can be avoided using idea called "ciphertext stealing" (as long as messages are more than 1 block)

interesting traffic analysis attack: each keystroke is sent in separate packet, so # packets leaks info on length of user's password!

Comparing CTR mode to CBC mode:

CTR mode	CBC mode
1. no padding needed (shorter ciphertexts)	1. padding needed
2. parallelizable	2. sequential
3. only requires PRF (no need to invert)	3. requires PRP
4. tighter security	4. less tight security (re-key more often)
5. IVs have to be non-repeating (and spaced far apart)	5. requires unpredictable IVs

imagine 1 byte messages (e.g., encrypted key strokes over SSH)

1 block + 1 byte with CTR
2 blocks with CBC

requires more structured primitive, more code to implement forward and backward evaluation

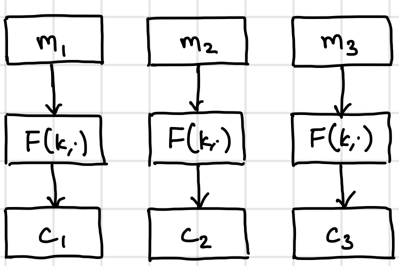
easy to implement: IV = nonce $\|$ counter

only needs to be non-repeating (can be predictable)

TLS 1.0 used predictable IVs (see HW1 for an attack)
SSH v1 used a 0 IV (even worse!)

Bottom-line: use randomized or nonce-based counter mode whenever possible: simpler, easier, and better than CBC!

A tempting and bad way to use a block cipher: ECB mode (electronic codebook)



Scheme is deterministic! Cannot be CPA secure!

Not even semantically secure!

(m_0, m_0) vs. (m_0, m_1) where $m_1 \neq m_0$

ciphertext blocks output are same

ciphertext blocks output are different

Encryption: simply apply block cipher to each block of the message

Decryption: simply invert each block of the ciphertext

NEVER USE ECB MODE FOR ENCRYPTION!