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

How do we reuse it? Choose a random starting point (called an initialization vector) in "randomized counter mode"

<table>
<thead>
<tr>
<th>IV</th>
<th>F(k,IV)</th>
<th>F(k,IV1)</th>
<th>F(k,IV2)</th>
<th>F(k,IV3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**decide message into blocks** (based on block size of PRF)

observe: ciphertext is longer than the message (required for CPA security)

**Theorem:** Let $F: K \times X \rightarrow Y$ be a secure PRF and let $T_{ctr}$ denote the randomized counter mode encryption scheme from above for $l$-block messages ($M = X^l$). Then, for all efficient CPA adversaries $A$, there exists an efficient PRF adversary $B$ such that

$$\text{CPAAdv}[A, T_{ctr}] \leq \frac{4Q^2l}{1X} + 2^{-E} \cdot \text{PRFAdv}[B, F]$$

$s$ : number of encryption queries

$l$ : number of blocks in message

**Intuition:**
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: $(X, X+1, \ldots, X+l-1)$ overlaps with $(X', X'+1, \ldots, X'+l-1)$ when $X, X' \in X$

$$\Pr\[\text{collision}\] \leq \frac{2lQ^2}{1X}$$

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

$$\Pr\[\text{collision}\] \leq \frac{2lQ^2}{1X}$$

3. Remaining factor of 2 in advantage due to intermediate distribution:

- Encrypt $m_0$ with PRF
- Encrypt $m_0$ with fresh one-time pad
- Encrypt $m_0$ with fresh one-time pad
- Encrypt $m_1$ with PRF

$$\Pr\[\text{PRFAdv}[B, F] + \frac{2lQ^2}{1X}\]$$

**Interpretation:** If $|X| = 2^{128}$ (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

$$Q \leq \sqrt{\frac{1X \cdot 2^{-32}}{4E}} = \sqrt{2^{16} \cdot 2^{-32}} = 2^{7/2} \approx 2^{39} \quad (\sim 1 \text{ trillion messages})$$
None-based counter mode: divide IV into two pieces: \( IV = \text{nonce} \| \text{counter} \)

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

Only requirement for security is that IV does not repeat:

- **Option 1:** Choose randomly (other 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

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

**Other block cipher modes of operation:**

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

Theorem: Let \( f : X \times K \rightarrow Y \) be a secure PRF and let \( \text{CBC} \) denote the CBC encryption scheme for \( l \)-block messages \( (M = X \times 2^l) \). Then, for all efficient CPA adversaries \( A \), there exists an efficient PRF adversary \( B \) such that

\[
\text{PRFAdv}[A, \text{CBC}] \leq 2^{\frac{2^l}{2}} + 2 \cdot \text{PRFAdv}[B, f]
\]

**Inuition:** Similar to analysis of randomized counter mode:

1. Ciphertext is indistinguishable from random string if PRF is evaluated on distinct inputs
2. When encrypting, PRF is invoked on \( l \) random blocks, so after \( Q \) queries, we have \( \Omega l \) random blocks.
   \[
   \Rightarrow \text{Collision probability} \leq 2^{\frac{2^l}{2}} \cdot \frac{2^l}{2^l} \leq \frac{2^l}{2} \cdot \frac{2^l}{2^l} \leq \frac{1}{2} \text{ [overlap of } \Omega \text{ random intervals vs. } \Omega l \text{ random points]}
   \]
3. Factor of 2 arises for same reason as before.

**Interpretation.** CBC mode provides weaker security compared to counter mode: \( 2^{2^l} \text{ vs. } 2^{2^l} \)

Concretely: for same parameters as before (1 MB messages, 2^12 distinguishing advantage):

\[
Q \leq \sqrt{ \frac{2^l}{2} \cdot 2^{31.5} } = 2^{\frac{2^l}{2}} = 2^l \cdot 2^{31.5} = 2^{31.5} \text{ (apart } 1 \text{ billion messages)}
\]

\[
\Rightarrow 2^{25} \approx 180 \times \text{smaller than using counter mode}
\]
Padding in CBC mode: each ciphertext block is computed by feeding a message block into the PRP
⇒ message must be an even multiple of the block size
⇒ 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: ]

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

Need to pad in CBC encryption can be exploited in "padding oracle" attacks — see HW1 for one example

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

Comparing CTR mode to CBC mode:

CTR mode
1. no padding needed (shorter ciphertexts)
2. parallelizable
3. only requires PRF (no need to invert)
4. tighter security
5. IVs have to be non-repeating (and spaced far apart)

CBC mode
1. padding needed
2. sequential
3. requires PRP
4. less tight security
5. requires unpredictable IVs

Easy to implement: IVs have to be non-repeating (can be predictable)

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)

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!