Cryptography

Lecture 9

Arkady Yerukhimovich

September 25, 2024

Outline

Lecture 8 Review

- Quiz on PRFs
- Started proof of CPA-security for PRF+OTP

Outline

1. Lecture 8 Review
2. Security of PRF+OTP (Chapter 3.5.2)
3. Modes of Operation (Chapter 3.6.2)

CPA-Secure Encryption from a PRF

PRF+OTP Encryption (Π)

- $\mathsf{Gen}(1^n)$: $k \leftarrow \{0,1\}^n$
- $\mathsf{Enc}(k,m)$: Choose $r \leftarrow \{0,1\}^n$, output $c = (r, F_k(r) \oplus m)$
- $\mathsf{Dec}(k,c)$: Parse c as (r, c'), compute $m = F_k(r) \oplus c'$

Theorem

If F is a secure PRF, then PRF+OTP is CPA-secure

Proof Technique

To prove security from a PRF, we often do the following:

1. Consider the scheme where F_k is replaced by a random function f
    - Show by reduction to security of PRF, that $\mathcal{A}$ can't tell we made this change.
    - So, $\mathcal{A}$'s success probability must be (essentially) the same in this and original variant.

Proof Technique

To prove security from a PRF, we often do the following:

1. Consider the scheme where F_k is replaced by a random function f
   - Show by reduction to security of PRF, that $\mathcal{A}$ can't tell we made this change.
   - So, $\mathcal{A}$'s success probability must be (essentially) the same in this and original variant.
2. Use a probabilistic argument to prove that scheme is unconditionally secure when using a random function f.
   - Random function is essentially a collection of 2^n OTPs
   - Proof is similar to proof of OTP, but need to account for probability of collision in r

Security of PRF+OTP: Step 1

Define the following encryption scheme $\tilde{\Pi}$:

$\tilde{\Pi}$ Encryption Scheme

- $\widetilde{\mathsf{Gen}}(1^n)$: $f \leftarrow \mathcal{F}_n$ (the set of functions $\{0,1\}^n \to \{0,1\}^n$)
- $\widetilde{\mathsf{Enc}}(k, m)$: Choose $r \leftarrow \{0,1\}^n$, output $c = (r, f(r) \oplus m)$
- $\widetilde{\mathsf{Dec}}(k, c)$: Parse c as (r, c'), compute $m = f(r) \oplus c'$

- Observe that this is exactly PRF+OTP with F_k replaced by f
- This encryption is not efficient as we cannot evaluate a random function
- But, it is useful as a "thought experiment" in the proof as it gives us a target for security

Security of PRF+OTP: Step 1

$\tilde{\Pi}$ Encryption Scheme

- $\widetilde{\mathsf{Gen}}(1^n)$: $f \leftarrow \mathcal{F}_n$ (the set of functions $\{0,1\}^n \rightarrow \{0,1\}^n$)
- $\widetilde{\mathsf{Enc}}(k, m)$: Choose $r \leftarrow \{0,1\}^n$, output $c = (r, f(r) \oplus m)$
- $\widetilde{\mathsf{Dec}}(k, c)$: Parse c as (r, c'), compute $m = f(r) \oplus c'$

Lemma: For any PPT $\mathcal{A}$

$$\left|\Pr[PrivK^{cpa}_{\mathcal{A},\Pi}(n) = 1] - \Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1]\right| \leq \mathsf{negl}(n)$$

A Story of Two Games

Security of PRF+OTP: Proof of Lemma

Lemma

For any PPT $\mathcal{A}$

$$\left|\Pr[PrivK^{cpa}_{\mathcal{A},\Pi}(n) = 1] - \Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1]\right| \leq \mathsf{negl}(n)$$

We prove this lemma by reduction:

Security of PRF+OTP: Proof of Lemma

Lemma

For any PPT $\mathcal{A}$

$$\left|\Pr[PrivK^{cpa}_{\mathcal{A},\Pi}(n) = 1] - \Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1]\right| \leq \mathsf{negl}(n)$$

We prove this lemma by reduction:

- Assume there is a PPT $\mathcal{A}_c$ that breaks this lemma

Security of PRF+OTP: Proof of Lemma

Lemma

For any PPT $\mathcal{A}$

$$\left|\Pr[PrivK^{cpa}_{\mathcal{A},\Pi}(n) = 1] - \Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1]\right| \leq \mathsf{negl}(n)$$

We prove this lemma by reduction:

- Assume there is a PPT $\mathcal{A}_c$ that breaks this lemma
  - $\mathcal{A}_c$ is a CPA-security adversary
  - What we care about is the difference in probability that $\mathcal{A}_c$ wins the CPA-security game when playing with Π vs. $\tilde{\Pi}$.

Security of PRF+OTP: Proof of Lemma

Lemma

For any PPT $\mathcal{A}$

$$\left|\Pr[PrivK^{cpa}_{\mathcal{A},\Pi}(n) = 1] - \Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1]\right| \leq \mathsf{negl}(n)$$

We prove this lemma by reduction:

- Assume there is a PPT $\mathcal{A}_c$ that breaks this lemma
  - $\mathcal{A}_c$ is a CPA-security adversary
  - What we care about is the difference in probability that $\mathcal{A}_c$ wins the CPA-security game when playing with Π vs. $\tilde{\Pi}$.
- Use this to construct $\mathcal{A}_r$ that breaks PRF security of F_k

The Two Adversaries

$PRF_{\mathcal{D},F}(n)$

- The challenger chooses $b \leftarrow \{0,1\}$.
  If $b = 0$, he chooses $f \leftarrow \mathcal{F}_n$ and gives $\mathcal{D}$ an oracle $\mathcal{O} = f$.
  if $b = 1$, he chooses $k \leftarrow \{0,1\}^n$, and gives $\mathcal{D}$ an oracle $\mathcal{O} = F_k$.
- With access to oracle $\mathcal{O}$, the distinguisher $\mathcal{D}$ outputs a bit b'
- $PRF_{\mathcal{D},F}(n) = 1$ (i.e., $\mathcal{D}$ wins) if $b' = b$

$\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

The Two Adversaries

$PRF_{\mathcal{D},F}(n)$

- The challenger chooses $b \leftarrow \{0,1\}$.
  If $b=0$, he chooses $f \leftarrow \mathcal{F}_n$ and gives $\mathcal{D}$ an oracle $\mathcal{O}=f$.
  if $b=1$, he chooses $k \leftarrow \{0,1\}^n$, and gives $\mathcal{D}$ an oracle $\mathcal{O}=F_k$.
- With access to oracle $\mathcal{O}$, the distinguisher $\mathcal{D}$ outputs a bit b'
- $PRF_{\mathcal{D},F}(n)=1$ (i.e., $\mathcal{D}$ wins) if $b'=b$

$\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0|=|m_1|$.
- The challenger chooses $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)=1$ (i.e., $\mathcal{A}$ wins) if $b'=b$.

We have to consider two adversaries, $\mathcal{A}_r$ and $\mathcal{A}_c$

The Two Adversaries

$PRF_{\mathcal{D},F}(n)$

- The challenger chooses $b \leftarrow \{0,1\}$.
  If $b = 0$, he chooses $f \leftarrow \mathcal{F}_n$ and gives $\mathcal{D}$ an oracle $\mathcal{O} = f$.
  if $b = 1$, he chooses $k \leftarrow \{0,1\}^n$, and gives $\mathcal{D}$ an oracle $\mathcal{O} = F_k$.
- With access to oracle $\mathcal{O}$, the distinguisher $\mathcal{D}$ outputs a bit b'
- $PRF_{\mathcal{D},F}(n) = 1$ (i.e., $\mathcal{D}$ wins) if $b' = b$

$\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

We have to consider two adversaries, $\mathcal{A}_r$ and $\mathcal{A}_c$

- The PRF adversary $\mathcal{A}_r$:

The Two Adversaries

$PRF_{\mathcal{D},F}(n)$

- The challenger chooses $b \leftarrow \{0,1\}$.
  If $b = 0$, he chooses $f \leftarrow \mathcal{F}_n$ and gives $\mathcal{D}$ an oracle $\mathcal{O} = f$.
  if $b = 1$, he chooses $k \leftarrow \{0,1\}^n$, and gives $\mathcal{D}$ an oracle $\mathcal{O} = F_k$.
- With access to oracle $\mathcal{O}$, the distinguisher $\mathcal{D}$ outputs a bit b'
- $PRF_{\mathcal{D},F}(n) = 1$ (i.e., $\mathcal{D}$ wins) if $b' = b$

$\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

We have to consider two adversaries, $\mathcal{A}_r$ and $\mathcal{A}_c$

- The PRF adversary $\mathcal{A}_r$:
  - $\mathcal{A}_r$ is playing the PRF security game

The Two Adversaries

$PRF_{\mathcal{D},F}(n)$

- The challenger chooses $b \leftarrow \{0,1\}$.
  If $b = 0$, he chooses $f \leftarrow \mathcal{F}_n$ and gives $\mathcal{D}$ an oracle $\mathcal{O} = f$.
  if $b = 1$, he chooses $k \leftarrow \{0,1\}^n$, and gives $\mathcal{D}$ an oracle $\mathcal{O} = F_k$.
- With access to oracle $\mathcal{O}$, the distinguisher $\mathcal{D}$ outputs a bit b'
- $PRF_{\mathcal{D},F}(n) = 1$ (i.e., $\mathcal{D}$ wins) if $b' = b$

$\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

We have to consider two adversaries, $\mathcal{A}_r$ and $\mathcal{A}_c$

- The PRF adversary $\mathcal{A}_r$:
  - $\mathcal{A}_r$ is playing the PRF security game
  - $\mathcal{A}_r$ is given oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$ where either $\mathcal{O} = F_k(\cdot)$ (a PRF), or $\mathcal{O} = f(\cdot)$ (a random function)

The Two Adversaries

$PRF_{\mathcal{D},F}(n)$

- The challenger chooses $b \leftarrow \{0,1\}$.
  If $b=0$, he chooses $f \leftarrow \mathcal{F}_n$ and gives $\mathcal{D}$ an oracle $\mathcal{O}=f$.
  if $b=1$, he chooses $k \leftarrow \{0,1\}^n$, and gives $\mathcal{D}$ an oracle $\mathcal{O}=F_k$.
- With access to oracle $\mathcal{O}$, the distinguisher $\mathcal{D}$ outputs a bit b'
- $PRF_{\mathcal{D},F}(n)=1$ (i.e., $\mathcal{D}$ wins) if $b'=b$

$\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0|=|m_1|$.
- The challenger chooses $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)=1$ (i.e., $\mathcal{A}$ wins) if $b'=b$.

We have to consider two adversaries, $\mathcal{A}_r$ and $\mathcal{A}_c$

- The PRF adversary $\mathcal{A}_r$:
  - $\mathcal{A}_r$ is playing the PRF security game
  - $\mathcal{A}_r$ is given oracle $\mathcal{O}:\{0,1\}^n \to \{0,1\}^n$ where either $\mathcal{O}=F_k(\cdot)$ (a PRF), or $\mathcal{O}=f(\cdot)$ (a random function)
- The CPA-security adversary $\mathcal{A}_c$ (who breaks the lemma):

The Two Adversaries

$PRF_{\mathcal{D},F}(n)$

- The challenger chooses $b \leftarrow \{0,1\}$.
  If $b = 0$, he chooses $f \leftarrow \mathcal{F}_n$ and gives $\mathcal{D}$ an oracle $\mathcal{O} = f$.
  if $b = 1$, he chooses $k \leftarrow \{0,1\}^n$, and gives $\mathcal{D}$ an oracle $\mathcal{O} = F_k$.
- With access to oracle $\mathcal{O}$, the distinguisher $\mathcal{D}$ outputs a bit b'
- $PRF_{\mathcal{D},F}(n) = 1$ (i.e., $\mathcal{D}$ wins) if $b' = b$

$\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

We have to consider two adversaries, $\mathcal{A}_r$ and $\mathcal{A}_c$

- The PRF adversary $\mathcal{A}_r$:
  - $\mathcal{A}_r$ is playing the PRF security game
  - $\mathcal{A}_r$ is given oracle $\mathcal{O} : \{0,1\}^n \rightarrow \{0,1\}^n$ where either $\mathcal{O} = F_k(\cdot)$ (a PRF), or $\mathcal{O} = f(\cdot)$ (a random function)
- The CPA-security adversary $\mathcal{A}_c$ (who breaks the lemma):
  - $\mathcal{A}_c$ plays the CPA-security game against either Π or $\tilde{\Pi}$

The Two Adversaries

$PRF_{\mathcal{D},F}(n)$

- The challenger chooses $b \leftarrow \{0,1\}$.
  If $b = 0$, he chooses $f \leftarrow \mathcal{F}_n$ and gives $\mathcal{D}$ an oracle $\mathcal{O} = f$.
  if $b = 1$, he chooses $k \leftarrow \{0,1\}^n$, and gives $\mathcal{D}$ an oracle $\mathcal{O} = F_k$.
- With access to oracle $\mathcal{O}$, the distinguisher $\mathcal{D}$ outputs a bit b'
- $PRF_{\mathcal{D},F}(n) = 1$ (i.e., $\mathcal{D}$ wins) if $b' = b$

$\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

We have to consider two adversaries, $\mathcal{A}_r$ and $\mathcal{A}_c$

- The PRF adversary $\mathcal{A}_r$:
  - $\mathcal{A}_r$ is playing the PRF security game
  - $\mathcal{A}_r$ is given oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$ where either $\mathcal{O} = F_k(\cdot)$ (a PRF), or $\mathcal{O} = f(\cdot)$ (a random function)
- The CPA-security adversary $\mathcal{A}_c$ (who breaks the lemma):
  - $\mathcal{A}_c$ plays the CPA-security game against either Π or $\tilde{\Pi}$
    - to answer encryption queries – The $\mathsf{Enc}(\cdot)$ oracle given to $\mathcal{A}_c$ in Π uses F_k and the oracle in $\tilde{\Pi}$ uses f

The Two Adversaries

$PRF_{\mathcal{D},F}(n)$

- The challenger chooses $b \leftarrow \{0,1\}$.
  If $b = 0$, he chooses $f \leftarrow \mathcal{F}_n$ and gives $\mathcal{D}$ an oracle $\mathcal{O} = f$.
  if $b = 1$, he chooses $k \leftarrow \{0,1\}^n$, and gives $\mathcal{D}$ an oracle $\mathcal{O} = F_k$.
- With access to oracle $\mathcal{O}$, the distinguisher $\mathcal{D}$ outputs a bit b'
- $PRF_{\mathcal{D},F}(n) = 1$ (i.e., $\mathcal{D}$ wins) if $b' = b$

$\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^n)$ outputs m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

We have to consider two adversaries, $\mathcal{A}_r$ and $\mathcal{A}_c$

- The PRF adversary $\mathcal{A}_r$:
  - $\mathcal{A}_r$ is playing the PRF security game
  - $\mathcal{A}_r$ is given oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$ where either $\mathcal{O} = F_k(\cdot)$ (a PRF), or $\mathcal{O} = f(\cdot)$ (a random function)
- The CPA-security adversary $\mathcal{A}_c$ (who breaks the lemma):
  - $\mathcal{A}_c$ plays the CPA-security game against either Π or $\tilde{\Pi}$
    - to answer encryption queries – The $\mathsf{Enc}(\cdot)$ oracle given to $\mathcal{A}_c$ in Π uses F_k and the oracle in $\tilde{\Pi}$ uses f
  - We care about the *difference* in $\mathcal{A}_c$'s WIN probability

Constructing $\mathcal{A}_r^{\mathcal{O}}$: Intuition

- $\mathcal{A}_r$ needs to use $\mathcal{A}_c$ to win PRF game

Constructing $\mathcal{A}_r^{\mathcal{O}}$: Intuition

- $\mathcal{A}_r$ needs to use $\mathcal{A}_c$ to win PRF game
- $\mathcal{A}_r$ acts as the challenger for $\mathcal{A}_c$ in CPA-security game

Constructing $\mathcal{A}_r^{\mathcal{O}}$: Intuition

- $\mathcal{A}_r$ needs to use $\mathcal{A}_c$ to win PRF game
- $\mathcal{A}_r$ acts as the challenger for $\mathcal{A}_c$ in CPA-security game
  - $\mathcal{A}_r$ must answer $\mathcal{A}_c$'s $\mathsf{Enc}(\cdot)$ queries (i.e., simulate the Enc oracle)
  - $\mathcal{A}_r$ must produce the challenge ciphertext c

Constructing $\mathcal{A}_r^{\mathcal{O}}$: Intuition

- $\mathcal{A}_r$ needs to use $\mathcal{A}_c$ to win PRF game
- $\mathcal{A}_r$ acts as the challenger for $\mathcal{A}_c$ in CPA-security game
  - $\mathcal{A}_r$ must answer $\mathcal{A}_c$'s $\mathsf{Enc}(\cdot)$ queries (i.e., simulate the Enc oracle)
  - $\mathcal{A}_r$ must produce the challenge ciphertext c
- If $\mathcal{A}_c$ WINS, $\mathcal{A}_r$ must use that to win the game against his challenger

Constructing $\mathcal{A}_r^{\mathcal{O}}$

1. Pre-Challenge
   Run $\mathcal{A}_c(1^n)$ and when $\mathcal{A}_c$ asks $\mathsf{Enc}(m)$ query
   - Choose $r \leftarrow \{0,1\}^n$, query $y = \mathcal{O}(r)$, return $c = (r, y \oplus m)$ to $\mathcal{A}_c$

Constructing $\mathcal{A}_r^{\mathcal{O}}$

1. Pre-Challenge
   Run $\mathcal{A}_c(1^n)$ and when $\mathcal{A}_c$ asks $\mathsf{Enc}(m)$ query
   - Choose $r \leftarrow \{0,1\}^n$, query $y = \mathcal{O}(r)$, return $c = (r, y \oplus m)$ to $\mathcal{A}_c$
2. Challenge
   When $\mathcal{A}_c$ outputs (m_0, m_1)
   - Choose $b \leftarrow \{0,1\}$
   - Choose $r \leftarrow \{0,1\}^n$, query $y = \mathcal{O}(r)$, return $c = (r, y \oplus m_b)$ as the challenge

Constructing $\mathcal{A}_r^{\mathcal{O}}$

1. Pre-Challenge
   Run $\mathcal{A}_c(1^n)$ and when $\mathcal{A}_c$ asks $\mathsf{Enc}(m)$ query
   - Choose $r \leftarrow \{0,1\}^n$, query $y = \mathcal{O}(r)$, return $c = (r, y \oplus m)$ to $\mathcal{A}_c$
2. Challenge
   When $\mathcal{A}_c$ outputs (m_0, m_1)
   - Choose $b \leftarrow \{0,1\}$
   - Choose $r \leftarrow \{0,1\}^n$, query $y = \mathcal{O}(r)$, return $c = (r, y \oplus m_b)$ as the challenge
3. Post-Challenge
   Continue answering Enc queries until $\mathcal{A}_c$ outputs guess b'
   - Output 1 ("PRF") if $b = b'$, and 0 otherwise.

Observation

- If $\mathcal{O}$ is f, then $\mathcal{A}_r$ is simulating $\tilde{\Pi}$
- If $\mathcal{O}$ if F_k, then $\mathcal{A}_r$ is simulating Π

Analysis of $\mathcal{A}_r$'s success

There are two cases to analyze:

Analysis of $\mathcal{A}_r$'s success

There are two cases to analyze:

- Case 1: $\mathcal{O} = F_k$ (i.e., $b = 1$ in PRF game)

Analysis of $\mathcal{A}_r$'s success

There are two cases to analyze:

- Case 1: $\mathcal{O} = F_k$ (i.e., $b = 1$ in PRF game)
  - $\mathcal{A}_r$ answers all Enc queries and produces c with $F_k(r) \oplus m$

Analysis of $\mathcal{A}_r$'s success

There are two cases to analyze:

- Case 1: $\mathcal{O} = F_k$ (i.e., $b = 1$ in PRF game)
  - $\mathcal{A}_r$ answers all Enc queries and produces c with $F_k(r) \oplus m$
  - This is exactly the CPA-security game vs. Π

Analysis of $\mathcal{A}_r$'s success

There are two cases to analyze:

- Case 1: $\mathcal{O} = F_k$ (i.e., $b = 1$ in PRF game)
    - $\mathcal{A}_r$ answers all Enc queries and produces c with $F_k(r) \oplus m$
    - This is exactly the CPA-security game vs. Π
    - Since $\mathcal{A}_r$ output 1 when $\mathcal{A}_c$ WINS, we have

$$\Pr_{k \leftarrow \{0,1\}^n}[\mathcal{A}_r^{F_k(\cdot)}(1^n) = 1] = \Pr[PrivK^{cpa}_{\mathcal{A}_c,\Pi}(n) = 1]$$

Analysis of $\mathcal{A}_r$'s success

There are two cases to analyze:

- Case 1: $\mathcal{O} = F_k$ (i.e., $b = 1$ in PRF game)
  - $\mathcal{A}_r$ answers all Enc queries and produces c with $F_k(r) \oplus m$
  - This is exactly the CPA-security game vs. Π
  - Since $\mathcal{A}_r$ output 1 when $\mathcal{A}_c$ WINS, we have

$$\Pr_{k \leftarrow \{0,1\}^n}[\mathcal{A}_r^{F_k(\cdot)}(1^n) = 1] = \Pr[PrivK^{cpa}_{\mathcal{A}_c,\Pi}(n) = 1]$$

- Case 2: $\mathcal{O} = f$ (i.e., $b = 0$ in PRF game)

Analysis of $\mathcal{A}_r$'s success

There are two cases to analyze:

- Case 1: $\mathcal{O} = F_k$ (i.e., $b = 1$ in PRF game)
  - $\mathcal{A}_r$ answers all Enc queries and produces c with $F_k(r) \oplus m$
  - This is exactly the CPA-security game vs. Π
  - Since $\mathcal{A}_r$ output 1 when $\mathcal{A}_c$ WINS, we have

$$\Pr_{k \leftarrow \{0,1\}^n}[\mathcal{A}_r^{F_k(\cdot)}(1^n) = 1] = \Pr[PrivK^{cpa}_{\mathcal{A}_c,\Pi}(n) = 1]$$

- Case 2: $\mathcal{O} = f$ (i.e., $b = 0$ in PRF game)
  - $\mathcal{A}_r$ answers all Enc queries and produces c with $f(r) \oplus m$

Analysis of $\mathcal{A}_r$'s success

There are two cases to analyze:

- Case 1: $\mathcal{O} = F_k$ (i.e., $b = 1$ in PRF game)
  - $\mathcal{A}_r$ answers all Enc queries and produces c with $F_k(r) \oplus m$
  - This is exactly the CPA-security game vs. Π
  - Since $\mathcal{A}_r$ output 1 when $\mathcal{A}_c$ WINS, we have

$$\Pr_{k \leftarrow \{0,1\}^n}[\mathcal{A}_r^{F_k(\cdot)}(1^n) = 1] = \Pr[PrivK_{\mathcal{A}_c,\Pi}^{cpa}(n) = 1]$$

- Case 2: $\mathcal{O} = f$ (i.e., $b = 0$ in PRF game)
  - $\mathcal{A}_r$ answers all Enc queries and produces c with $f(r) \oplus m$
  - This is exactly the CPA-security game vs. $\tilde{\Pi}$

Analysis of $\mathcal{A}_r$'s success

There are two cases to analyze:

- Case 1: $\mathcal{O} = F_k$ (i.e., $b = 1$ in PRF game)
  - $\mathcal{A}_r$ answers all Enc queries and produces c with $F_k(r) \oplus m$
  - This is exactly the CPA-security game vs. Π
  - Since $\mathcal{A}_r$ output 1 when $\mathcal{A}_c$ WINS, we have

$$\Pr_{k \leftarrow \{0,1\}^n}[\mathcal{A}_r^{F_k(\cdot)}(1^n) = 1] = \Pr[PrivK^{cpa}_{\mathcal{A}_c,\Pi}(n) = 1]$$

- Case 2: $\mathcal{O} = f$ (i.e., $b = 0$ in PRF game)
  - $\mathcal{A}_r$ answers all Enc queries and produces c with $f(r) \oplus m$
  - This is exactly the CPA-security game vs. $\tilde{\Pi}$
  - Since $\mathcal{A}_r$ output 1 when $\mathcal{A}_c$ WINS, we have

$$\Pr_{f \leftarrow \mathcal{F}_n}[\mathcal{A}_r^{f(\cdot)}(1^n) = 1] = \Pr[PrivK^{cpa}_{\mathcal{A}_c,\tilde{\Pi}}(n) = 1]$$

Analysis of $\mathcal{A}_r$'s success

- We assumed that $\mathcal{A}_c$ breaks the lemma – i.e. has different success probability vs. Π and $\tilde{\Pi}$

$$\left|\Pr[PrivK^{cpa}_{\mathcal{A}_c,\Pi}(n) = 1] - \Pr[PrivK^{cpa}_{\mathcal{A}_c,\tilde{\Pi}}(n) = 1]\right| > 1/\text{poly}(n)$$

Analysis of $\mathcal{A}_r$'s success

- We assumed that $\mathcal{A}_c$ breaks the lemma – i.e. has different success probability vs. Π and $\tilde{\Pi}$

$$\left|\Pr[PrivK^{cpa}_{\mathcal{A}_c,\Pi}(n) = 1] - \Pr[PrivK^{cpa}_{\mathcal{A}_c,\tilde{\Pi}}(n) = 1]\right| > 1/\text{poly}(n)$$

- By the last slide, this implies that

$$\left|\Pr_{k \leftarrow \{0,1\}^n}[\mathcal{A}_r^{F_k(\cdot)}(1^n) = 1] - \Pr_{f \leftarrow \mathcal{F}_n}[\mathcal{A}_r^{f(\cdot)}(1^n) = 1]\right| > 1/\text{poly}(n)$$

Analysis of $\mathcal{A}_r$'s success

- We assumed that $\mathcal{A}_c$ breaks the lemma – i.e. has different success probability vs. Π and $\tilde{\Pi}$

$$\left|\Pr[PrivK^{cpa}_{\mathcal{A}_c,\Pi}(n) = 1] - \Pr[PrivK^{cpa}_{\mathcal{A}_c,\tilde{\Pi}}(n) = 1]\right| > 1/\text{poly}(n)$$

- By the last slide, this implies that

$$\left|\Pr_{k\leftarrow\{0,1\}^n}[\mathcal{A}_r^{F_k(\cdot)}(1^n) = 1] - \Pr_{f\leftarrow\mathcal{F}_n}[\mathcal{A}_r^{f(\cdot)}(1^n) = 1]\right| > 1/\text{poly}(n)$$

- That is, $\mathcal{A}_r$ is able to distinguish between $F_k(\cdot)$ and $f(\cdot)$. But, we know that F_k is a PRF.

Contradiction!

Proof Technique

To prove security from a PRF, we often do the following:

✓ Consider the scheme where F_k is replaced by a random function f
  - Show by reduction to security of PRF, that $\mathcal{A}$ can't tell we made this change.
  - So, $\mathcal{A}$'s success probability must be (essentially) the same in this and original variant.

2. Use a probabilistic argument to prove that scheme is unconditionally secure when using a random function f.
   - Random function is essentially a collection of 2^n OTPs
   - Proof is similar to proof of OTP, but need to account for probability of collision in r

Proving CPA-security of $\tilde{\Pi}$

Lemma

For any $\mathcal{A}$ making at most $q(n)$ queries to $\mathrm{Enc}(\cdot)$

$$\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \leq 1/2 + \frac{q(n)}{2^n}$$

Proving CPA-security of $\tilde{\Pi}$

Lemma

For any $\mathcal{A}$ making at most $q(n)$ queries to $\mathrm{Enc}(\cdot)$

$$\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \leq 1/2 + \frac{q(n)}{2^n}$$

- Recall that $\tilde{\Pi}$ encrypts as $c = (r, f(r) \oplus m)$

Proving CPA-security of $\tilde{\Pi}$

Lemma

For any $\mathcal{A}$ making at most $q(n)$ queries to $\mathrm{Enc}(\cdot)$

$$\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \leq 1/2 + \frac{q(n)}{2^n}$$

- Recall that $\tilde{\Pi}$ encrypts as $c = (r, f(r) \oplus m)$
- Let r^* be the randomness used to encrypt the challenge

Proving CPA-security of $\tilde{\Pi}$

Lemma

For any $\mathcal{A}$ making at most $q(n)$ queries to $\mathrm{Enc}(\cdot)$

$$\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \leq 1/2 + \frac{q(n)}{2^n}$$

- Recall that $\tilde{\Pi}$ encrypts as $c = (r, f(r) \oplus m)$
- Let r^* be the randomness used to encrypt the challenge
- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries

Proving CPA-security of $\tilde{\Pi}$

Lemma

For any $\mathcal{A}$ making at most $q(n)$ queries to $\mathrm{Enc}(\cdot)$

$$\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \leq 1/2 + \frac{q(n)}{2^n}$$

- Recall that $\tilde{\Pi}$ encrypts as $c = (r, f(r) \oplus m)$
- Let r^* be the randomness used to encrypt the challenge
- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\mathcal{A}$ knows nothing about $f(r^*)$, so $f(r^*)$ is random (good OTP)
  - $\Pr[\mathcal{A}$ outputs $b = b'] = 1/2$

Proving CPA-security of $\tilde{\Pi}$

Lemma

For any $\mathcal{A}$ making at most $q(n)$ queries to $\mathrm{Enc}(\cdot)$

$$\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \leq 1/2 + \frac{q(n)}{2^n}$$

- Recall that $\tilde{\Pi}$ encrypts as $c = (r, f(r) \oplus m)$
- Let r^* be the randomness used to encrypt the challenge
- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\mathcal{A}$ knows nothing about $f(r^*)$, so $f(r^*)$ is random (good OTP)
  - $\Pr[\mathcal{A}$ outputs $b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries

Proving CPA-security of $\tilde{\Pi}$

Lemma

For any $\mathcal{A}$ making at most $q(n)$ queries to $\mathrm{Enc}(\cdot)$

$$\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \leq 1/2 + \frac{q(n)}{2^n}$$

- Recall that $\tilde{\Pi}$ encrypts as $c = (r, f(r) \oplus m)$
- Let r^* be the randomness used to encrypt the challenge
- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\mathcal{A}$ knows nothing about $f(r^*)$, so $f(r^*)$ is random (good OTP)
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\mathcal{A}$ learns value of $f(r^*)$ (he sees $c = (r^*, c')$, computes $f(r^*) = c' \oplus m$)
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$

Proving CPA-security of $\tilde{\Pi}$

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$

Proving CPA-security of $\tilde{\Pi}$

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$

Claim: $\Pr[\text{Case 2}] \leq \text{negl}(n)$

Proving CPA-security of $\tilde{\Pi}$

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$

Claim: $\Pr[\text{Case 2}] \leq \mathsf{negl}(n)$

- We said that $\mathcal{A}$ makes at most $q(n) = \mathsf{poly}(n)$ Enc queries

Proving CPA-security of $\tilde{\Pi}$

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$

Claim: $\Pr[\text{Case 2}] \leq \text{negl}(n)$

- We said that $\mathcal{A}$ makes at most $q(n) = \text{poly}(n)$ Enc queries
- On each Enc query, randomness $r_i \leftarrow \{0,1\}^n$

Proving CPA-security of $\tilde{\Pi}$

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$

Claim: $\Pr[\text{Case } 2] \leq \mathsf{negl}(n)$

- We said that $\mathcal{A}$ makes at most $q(n) = \mathsf{poly}(n)$ Enc queries
- On each Enc query, randomness $r_i \leftarrow \{0,1\}^n$
- In encrypting challenge, $r^* \leftarrow \{0,1\}^n$

Proving CPA-security of $\tilde{\Pi}$

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$

Claim: $\Pr[\text{Case 2}] \leq \mathsf{negl}(n)$

- We said that $\mathcal{A}$ makes at most $q(n) = \mathsf{poly}(n)$ Enc queries
- On each Enc query, randomness $r_i \leftarrow \{0,1\}^n$
- In encrypting challenge, $r^* \leftarrow \{0,1\}^n$
- So,

$$\Pr[r^* \in \{r_1, \ldots, r_{q(n)}\}] \leq \sum_{i=1}^{q(n)} \Pr[r^* = r_i] = \frac{q(n)}{2^n} \leq \mathsf{negl}(n)$$

Proving CPA-security of $\tilde{\Pi}$: Putting It Together

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$
  - Occurs with probability at most $q(n)/2^n$

Proving CPA-security of $\tilde{\Pi}$: Putting It Together

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$
  - Occurs with probability at most $q(n)/2^n$

$$\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] \;=\; \Pr[\mathcal{A} \text{ WINS } \wedge \text{ Case 1}] + \Pr[\mathcal{A} \text{ WINS } \wedge \text{ Case 2}]$$

Proving CPA-security of $\tilde{\Pi}$: Putting It Together

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$
  - Occurs with probability at most $q(n)/2^n$

$$\begin{aligned}
\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] &= \Pr[\mathcal{A} \text{ WINS } \wedge \text{ Case 1}] + \Pr[\mathcal{A} \text{ WINS } \wedge \text{ Case 2}] \\
&\leq \Pr[\mathcal{A} \text{ WINS} \mid \text{Case 1}] \cdot \Pr[\text{Case 1}] + \Pr[\text{Case 2}]
\end{aligned}$$

Proving CPA-security of $\tilde{\Pi}$: Putting It Together

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$
  - Occurs with probability at most $q(n)/2^n$

$$
\begin{aligned}
\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] &= \Pr[\mathcal{A} \text{ WINS } \wedge \text{ Case 1}] + \Pr[\mathcal{A} \text{ WINS } \wedge \text{ Case 2}] \\
&\leq \Pr[\mathcal{A} \text{ WINS} \mid \text{Case 1}] \cdot \Pr[\text{Case 1}] + \Pr[\text{Case 2}] \\
&\leq \Pr[\mathcal{A} \text{ WINS} \mid \text{Case 1}] + \Pr[\text{Case 2}]
\end{aligned}
$$

Proving CPA-security of $\tilde{\Pi}$: Putting It Together

- Case 1: r^* is never used when answering $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1/2$
- Case 2: r^* is used when answering one of $\mathcal{A}$'s Enc queries
  - $\Pr[\mathcal{A} \text{ outputs } b = b'] = 1$
  - Occurs with probability at most $q(n)/2^n$

$$
\begin{aligned}
\Pr[PrivK^{cpa}_{\mathcal{A},\tilde{\Pi}}(n) = 1] &= \Pr[\mathcal{A} \text{ WINS } \wedge \text{ Case 1}] + \Pr[\mathcal{A} \text{ WINS } \wedge \text{ Case 2}] \\
&\le \Pr[\mathcal{A} \text{ WINS} \mid \text{Case 1}] \cdot \Pr[\text{Case 1}] + \Pr[\text{Case 2}] \\
&\le \Pr[\mathcal{A} \text{ WINS} \mid \text{Case 1}] + \Pr[\text{Case 2}] \\
&\le 1/2 + \frac{q(n)}{2^n}
\end{aligned}
$$

Finishing Proof of CPA-security of PRF+OTP

- ✓ Consider the scheme where F_k is replaced by a random function f
  - We showed that any PPT $\mathcal{A}$ has only a $\text{negl}(n)$ advantage in distinguishing the two games
- ✓ Use a probabilistic argument to prove that scheme is unconditionally secure when using a random function f.
  - We showed that PPT $\mathcal{A}$ WINS with probability $\leq 1/2 + q(n)/2^n$

Finishing Proof of CPA-security of PRF+OTP

- ✓ Consider the scheme where F_k is replaced by a random function f
  - We showed that any PPT $\mathcal{A}$ has only a $\text{negl}(n)$ advantage in distinguishing the two games
- ✓ Use a probabilistic argument to prove that scheme is unconditionally secure when using a random function f.
  - We showed that PPT $\mathcal{A}$ WINS with probability $\leq 1/2 + q(n)/2^n$

Combining these two statements, we get that for any PPT $\mathcal{A}$,

$$\Pr[PrivK^{cpa}_{\mathcal{A},\text{PRF+OTP}}(n) = 1] \leq 1/2 + \frac{q(n)}{2^n} + \text{negl}(n)$$

Outline

Encrypting Long Messages

CPA-secure encryption allows us to

- Encrypt many messages using the same key k

Encrypting Long Messages

CPA-secure encryption allows us to

- Encrypt many messages using the same key k
- Encrypt messages longer than the key k

Encrypting Long Messages

CPA-secure encryption allows us to

- Encrypt many messages using the same key k
- Encrypt messages longer than the key k

Are we done?

Encrypting Long Messages

Consider how we encrypt long messages:

Encrypting Long Messages

Consider how we encrypt long messages:

- Break m into blocks $m_1, \ldots, m_\ell$ such that $|m_i| = n$
- Encrypt each block separately using key k.

Encrypting Long Messages

Consider how we encrypt long messages:

- Break m into blocks $m_1, \ldots, m_\ell$ such that $|m_i| = n$
- Encrypt each block separately using key k.

Now, suppose we do this using PRF+OTP encryption

Encrypting Long Messages

Consider how we encrypt long messages:

- Break m into blocks $m_1, \ldots, m_\ell$ such that $|m_i| = n$
- Encrypt each block separately using key k.

Now, suppose we do this using PRF+OTP encryption

$$\begin{aligned} \mathsf{Enc}_k(m) &= \mathsf{Enc}_k(m_1), \mathsf{Enc}_k(m_2), \ldots, \mathsf{Enc}_k(m_\ell) \\ &= (r_1, F_k(r_1) \oplus m_1), (r_2, F_k(r_2) \oplus m_2), \ldots, (r_\ell, F_k(r_\ell) \oplus m_\ell) \end{aligned}$$

Encrypting Long Messages

Consider how we encrypt long messages:

- Break m into blocks $m_1, \ldots, m_\ell$ such that $|m_i| = n$
- Encrypt each block separately using key k.

Now, suppose we do this using PRF+OTP encryption

$$\begin{aligned}
\mathsf{Enc}_k(m) &= \mathsf{Enc}_k(m_1), \mathsf{Enc}_k(m_2), \ldots, \mathsf{Enc}_k(m_\ell) \\
&= (r_1, F_k(r_1) \oplus m_1), (r_2, F_k(r_2) \oplus m_2), \ldots, (r_\ell, F_k(r_\ell) \oplus m_\ell)
\end{aligned}$$

The problem:

Encrypting Long Messages

Consider how we encrypt long messages:

- Break m into blocks $m_1, \ldots, m_\ell$ such that $|m_i| = n$
- Encrypt each block separately using key k.

Now, suppose we do this using PRF+OTP encryption

$$\begin{aligned}\mathsf{Enc}_k(m) &= \mathsf{Enc}_k(m_1), \mathsf{Enc}_k(m_2), \ldots, \mathsf{Enc}_k(m_\ell) \\ &= (r_1, F_k(r_1) \oplus m_1), (r_2, F_k(r_2) \oplus m_2), \ldots, (r_\ell, F_k(r_\ell) \oplus m_\ell)\end{aligned}$$

The problem:

- $|c| = 2|m|$ (we have doubled the length of plaintext)

Encrypting Long Messages

Consider how we encrypt long messages:

- Break m into blocks $m_1, \ldots, m_\ell$ such that $|m_i| = n$
- Encrypt each block separately using key k.

Now, suppose we do this using PRF+OTP encryption

$$\begin{aligned}\mathsf{Enc}_k(m) &= \mathsf{Enc}_k(m_1), \mathsf{Enc}_k(m_2), \ldots, \mathsf{Enc}_k(m_\ell) \\ &= (r_1, F_k(r_1) \oplus m_1), (r_2, F_k(r_2) \oplus m_2), \ldots, (r_\ell, F_k(r_\ell) \oplus m_\ell)\end{aligned}$$

The problem:

- $|c| = 2|m|$ (we have doubled the length of plaintext)
- This is very problematic when sending big files (e.g., movies) or for HD encryption

Encrypting Long Messages

Consider how we encrypt long messages:

- Break m into blocks $m_1, \ldots, m_\ell$ such that $|m_i| = n$
- Encrypt each block separately using key k.

Now, suppose we do this using PRF+OTP encryption

$$\begin{aligned} \mathsf{Enc}_k(m) &= \mathsf{Enc}_k(m_1), \mathsf{Enc}_k(m_2), \ldots, \mathsf{Enc}_k(m_\ell) \\ &= (r_1, F_k(r_1) \oplus m_1), (r_2, F_k(r_2) \oplus m_2), \ldots, (r_\ell, F_k(r_\ell) \oplus m_\ell) \end{aligned}$$

The problem:

- $|c| = 2|m|$ (we have doubled the length of plaintext)
- This is very problematic when sending big files (e.g., movies) or for HD encryption

The Question

How do we encrypt long messages without this 2X increase in size

Block-Cipher Modes of Operations

- Modes of operation study how to encrypt many block messages without blow-up in size

Block-Cipher Modes of Operations

- Modes of operation study how to encrypt many block messages without blow-up in size
- Combine PRFs, Boolean operations, and randomness to achieve this

Block-Cipher Modes of Operations

- Modes of operation study how to encrypt many block messages without blow-up in size
- Combine PRFs, Boolean operations, and randomness to achieve this
- Can think of them as other ways to turn PRF into CPA-secure encryption

Modes of Operations: Important Properties

- Ciphertext length – what is the increase between $|c|$ and $|m|$

Modes of Operations: Important Properties

- Ciphertext length – what is the increase between $|c|$ and $|m|$
- Pre-computation – can some part of encryption procedure be computed without m

Modes of Operations: Important Properties

- Ciphertext length – what is the increase between $|c|$ and $|m|$
- Pre-computation – can some part of encryption procedure be computed without m
- Encryption/Decryption parallelism - can we encrypt/decrypt only part of m, or encrypt blocks in parallel

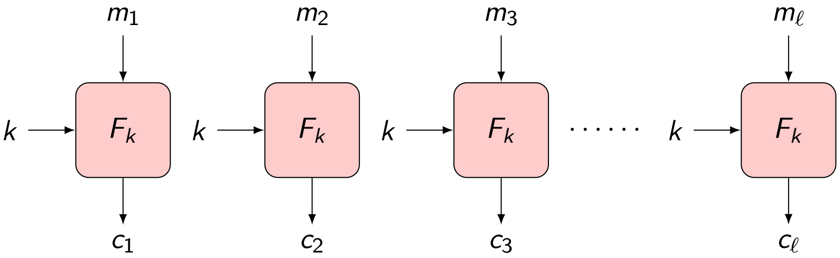
Modes of Operations: Important Properties

- Ciphertext length – what is the increase between $|c|$ and $|m|$
- Pre-computation – can some part of encryption procedure be computed without m
- Encryption/Decryption parallelism - can we encrypt/decrypt only part of m, or encrypt blocks in parallel
- PRF Inverse – do we need to be able to invert the PRF (i.e., do we need a PRP)

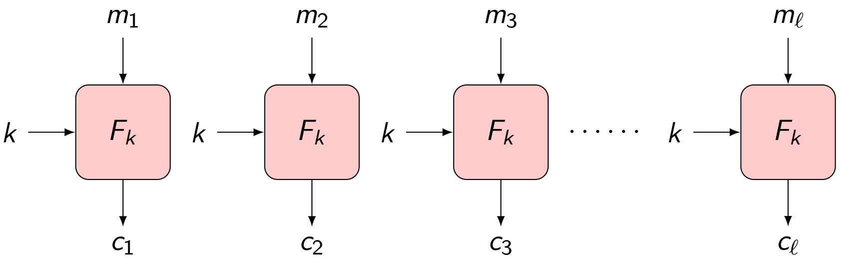
Modes of Operations: Important Properties

- Ciphertext length – what is the increase between $|c|$ and $|m|$
- Pre-computation – can some part of encryption procedure be computed without m
- Encryption/Decryption parallelism - can we encrypt/decrypt only part of m, or encrypt blocks in parallel
- PRF Inverse – do we need to be able to invert the PRF (i.e., do we need a PRP)
- Security - want at least CPA-security

Electronic Code Book (ECB) Mode

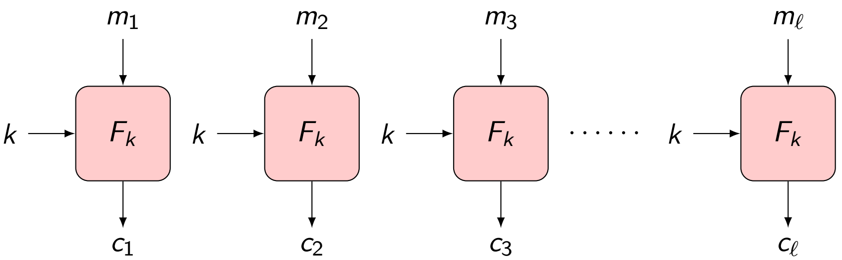


Electronic Code Book (ECB) Mode



- Ciphertext length: $|c| = |m|$
- Pre-computation: N/A
- Parallelism: Can encrypt/decrypt blocks in parallel
- PRF Inverse: Need to compute inverse to decrypt
- Security: ??

Electronic Code Book (ECB) Mode



- Ciphertext length: $|c| = |m|$
- Pre-computation: N/A
- Parallelism: Can encrypt/decrypt blocks in parallel
- PRF Inverse: Need to compute inverse to decrypt
- Security: NOT SECURE
  - ECB Mode is deterministic
  - Can tell if two blocks are the same

Electronic Code Book (ECB) Mode: How bad is it?

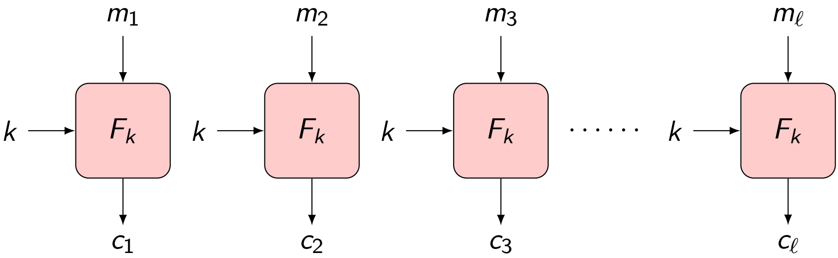


Figure: Original Image

Electronic Code Book (ECB) Mode: How bad is it?

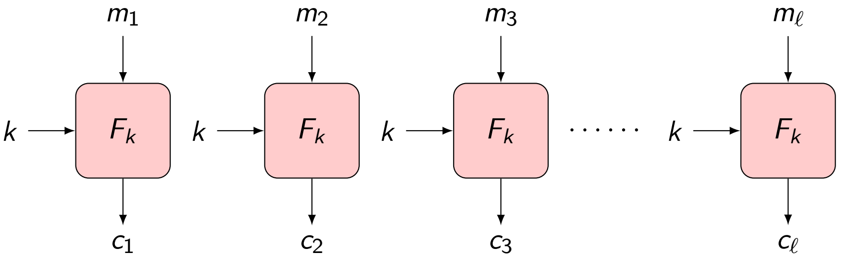


Figure: Original Image

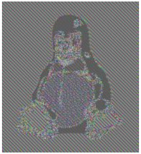


Figure: ECB-encrypted

Electronic Code Book (ECB) Mode: How bad is it?

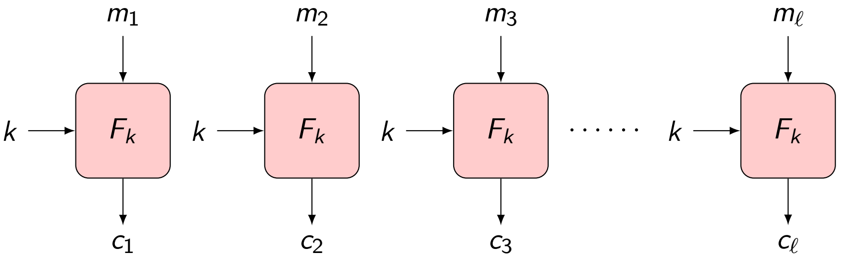


Figure: Original Image

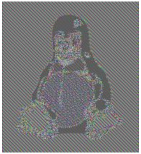
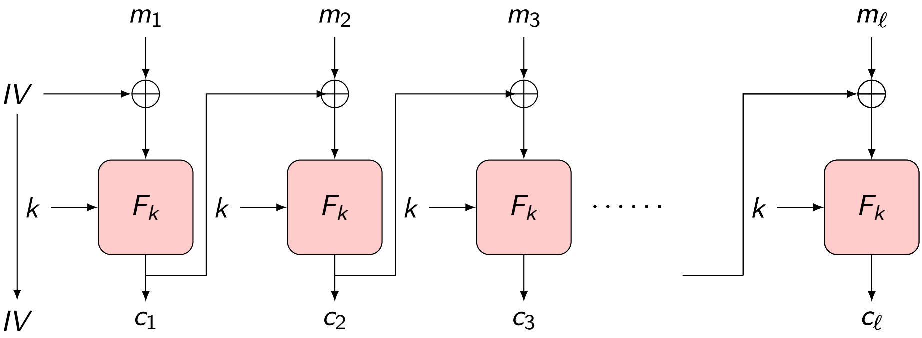


Figure: ECB-encrypted

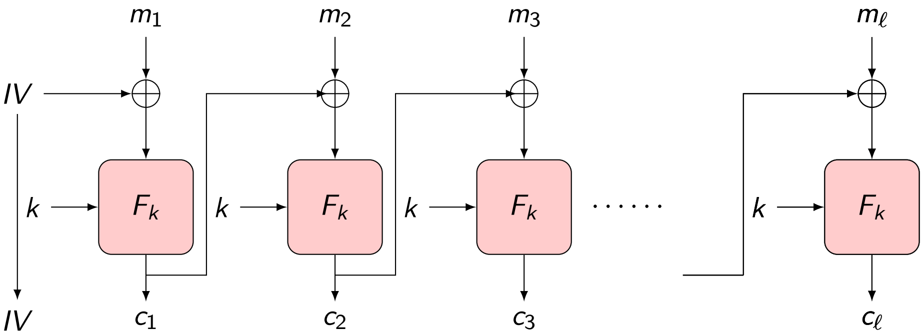
Warning

Never use ECB mode

Cipher Block Chaining (CBC) Mode

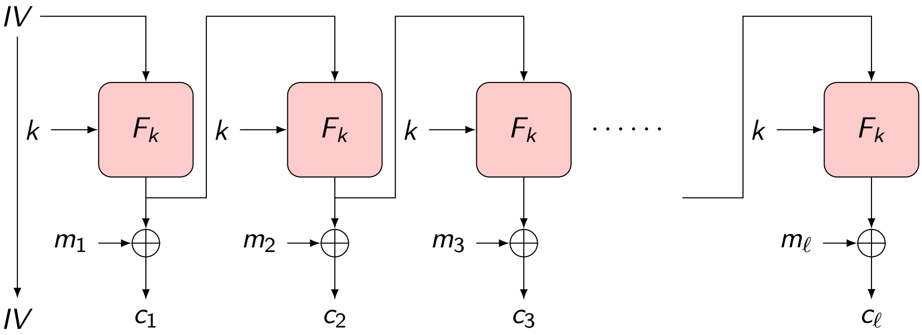


Cipher Block Chaining (CBC) Mode

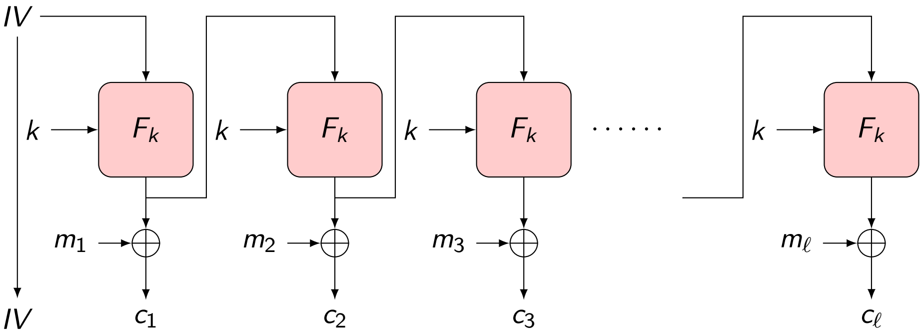


- Ciphertext length: $|c| = |m| + 1$ blocks
- Pre-computation: N/A
- Parallelism: Encryption / Decryption sequential
- PRF inverse: Need to compute inverse to decrypt
- Security: CPA-secure

Output Feedback (OFB) Mode

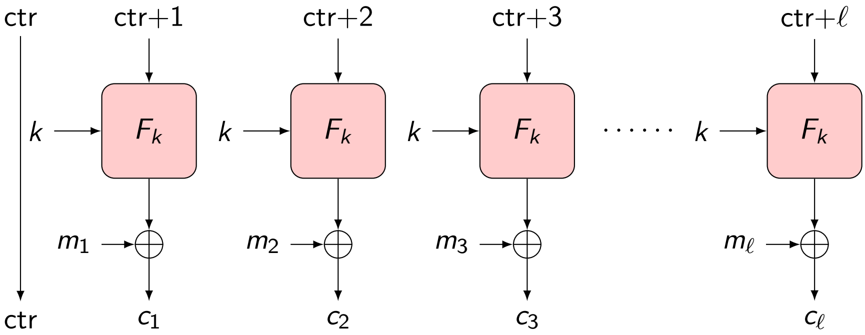


Output Feedback (OFB) Mode

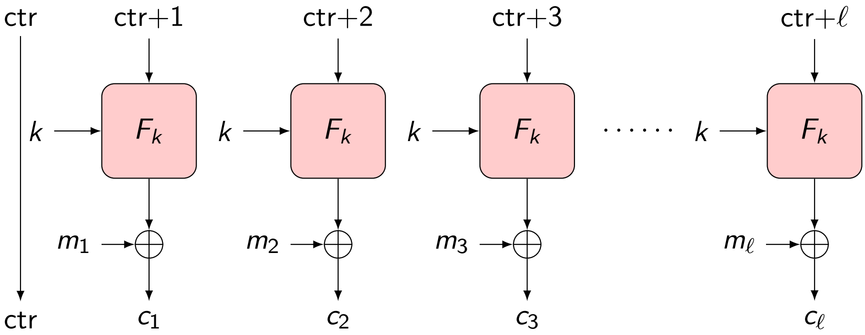


- Ciphertext length: $|c| = |m| + 1$ blocks
- Pre-computation: Can pre-compute entire pad
- Parallelism: If have pad, all encryption/decryption can be parallel
- PRF inverse: No need to compute inverse to decrypt
- Security: CPA-secure

Counter (CTR) Mode



Counter (CTR) Mode



- Ciphertext length: $|c| = |m| + 1$ blocks
- Pre-computation: Can pre-compute entire pad (or any part of pad)
- Parallelism: Can encrypt/decrypt any blocks in parallel
- PRF inverse: No need to compute inverse to decrypt
- Security: CPA-secure