CPA-secure encryption from a PRF:

$\Sigma[F]$:

$$\mathcal{K} = \{0,1\}^\lambda$$
$$\mathcal{M} = \{0,1\}^{\text{out}}$$
$$\mathcal{C} = \{0,1\}^\lambda \times \{0,1\}^{\text{out}}$$

$\underline{\text{KeyGen}:}$
$\quad k \leftarrow \{0,1\}^\lambda$
$\quad$ return k

$\underline{\text{Enc}(k,m):}$
$\quad r \leftarrow \{0,1\}^\lambda$
$\quad x := F(k,r) \oplus m$
$\quad$ return (r,x)

$\underline{\text{Dec}(k,(r,x)):}$
$\quad m := F(k,r) \oplus x$
$\quad$ return m

Claim:

If F is a secure PRF (with $\text{in} = \lambda$) then Σ is a CPA\$-secure encryption scheme. That is, $\mathcal{L}^{\Sigma}_{\text{cpa\$-real}} \approx \mathcal{L}^{\Sigma}_{\text{cpa\$-rand}}$.

Overview:

Want to show:

$$\begin{array}{|l|}\hline \mathcal{L}^{\Sigma}_{\text{cpa\$-real}} \\ \hline k \leftarrow \{0,1\}^\lambda \\ \\ \underline{\textsc{challenge}(m):} \\ \quad r \leftarrow \{0,1\}^\lambda \\ \quad x := F(k,r) \oplus m \\ \quad \text{return } (r,x) \\ \hline \end{array} \approx \begin{array}{|l|}\hline \mathcal{L}^{\Sigma}_{\text{cpa\$-rand}} \\ \hline \underline{\textsc{query}():} \\ \quad c \leftarrow \{0,1\}^{\lambda+\text{out}} \\ \quad \text{return } c \\ \hline \end{array}$$

The proof will **use** the fact F is a secure PRF. In other words,

$$\begin{array}{|l|}\hline \mathcal{L}^{F}_{\text{prf-real}} \\ \hline k \leftarrow \{0,1\}^\lambda \\ \\ \underline{\textsc{query}(r):} \\ \quad \text{return } F(k,r) \\ \hline \end{array} \approx \begin{array}{|l|}\hline \mathcal{L}^{F}_{\text{prf-rand}} \\ \hline T := \text{empty} \\ \\ \underline{\textsc{query}(r):} \\ \quad \text{if } T[r] \text{ undefined:} \\ \qquad T[r] \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } T[r] \\ \hline \end{array}$$

$$\mathcal{L}^{\Sigma}_{\text{cpa\$-real}}$$

$k \leftarrow \{0,1\}^\lambda$

$\underline{\textsc{challenge}(m):}$
  $r \leftarrow \{0,1\}^\lambda$
  $x := F(k,r) \oplus m$
  return (r,x)

Starting point is $\mathcal{L}^{\Sigma}_{\text{cpa\$-real}}$.

Security proof

$\mathcal{L}^{\Sigma}_{\text{cpa\$-real}}$

$k \leftarrow \{0,1\}^\lambda$

$\underline{\textsc{challenge}(m):}$

$r \leftarrow \{0,1\}^\lambda$
$x := F(k,r) \oplus m$
return (r, x)

Starting point is $\mathcal{L}^{\Sigma}_{\text{cpa\$-real}}$. Factor out call to F.

Security proof

$\underline{\textsc{challenge}(m)\text{:}}$
$r \leftarrow \{0,1\}^\lambda$
$z := \textsc{query}(r)$
$x := z \oplus m$
return (r,x)

$\diamond$

$\mathcal{L}^F_{\text{prf-real}}$
$k \leftarrow \{0,1\}^\lambda$
$\underline{\textsc{query}(r)\text{:}}$
return $F(k,r)$

Starting point is $\mathcal{L}^{\Sigma}_{\text{cpa\$-real}}$. Factor out call to F.

Security proof

$$
\boxed{\begin{array}{l}
\underline{\textsc{challenge}(m):} \\
\quad r \leftarrow \{0,1\}^\lambda \\
\quad z := \textsc{query}(r) \\
\quad x := z \oplus m \\
\quad \text{return } (r, x)
\end{array}}
\diamond
\boxed{\begin{array}{l}
\multicolumn{1}{c}{\mathcal{L}^F_{\text{prf-real}}} \\
\hline
k \leftarrow \{0,1\}^\lambda \\
\\
\underline{\textsc{query}(r):} \\
\quad \text{return } F(k, r)
\end{array}}
$$

Starting point is $\mathcal{L}^\Sigma_{\text{cpa\$-real}}$. Factor out call to F.

Security proof

$$\boxed{\begin{array}{l} \underline{\text{CHALLENGE}(m):} \\ \quad r \leftarrow \{0,1\}^\lambda \\ \quad z := \text{QUERY}(r) \\ \quad x := z \oplus m \\ \quad \text{return } (r, x) \end{array}} \diamond \boxed{\begin{array}{l} \mathcal{L}^F_{\text{prf-rand}} \\ \hline T := \text{empty} \\ \\ \underline{\text{QUERY}(r):} \\ \quad \text{if } T[r] \text{ undefined:} \\ \qquad T[r] \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } T[r] \end{array}}$$

Apply security of F: replace $\mathcal{L}_{\text{prf-real}}$ with $\mathcal{L}_{\text{prf-rand}}$.

$$\begin{array}{|l|}\hline \underline{\textsc{challenge}(m):} \\ \quad r \leftarrow \{0,1\}^\lambda \\ \quad z := \textsc{query}(r) \\ \quad x := z \oplus m \\ \quad \text{return } (r, x) \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}^F_{\text{prf-rand}} \\ \hline T := \text{empty} \\ \\ \underline{\textsc{query}(r):} \\ \quad \text{if } T[r] \text{ undefined:} \\ \qquad T[r] \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } T[r] \\ \hline \end{array}$$

Apply security of F: replace $\mathcal{L}_{\text{prf-real}}$ with $\mathcal{L}_{\text{prf-rand}}$. **Are we done?**

Security proof

CHALLENGE(m):		$\mathcal{L}^{F}_{\text{prf-rand}}$
$r \leftarrow \{0,1\}^{\lambda}$ $z := \text{QUERY}(r)$ $x := z \oplus m$ return (r, x)	$\diamond$	$T :=$ empty QUERY(r): if $T[r]$ undefined: $T[r] \leftarrow \{0,1\}^{\text{out}}$ return $T[r]$

If r happens to repeat (which is possible), one-time pad z is reused!

Security proof

$$\begin{array}{|l|}\hline \underline{\textsc{challenge}(m):} \\ \quad r \leftarrow \{0,1\}^\lambda \\ \quad z := \textsc{query}(r) \\ \quad x := z \oplus m \\ \quad \text{return } (r, x) \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}^F_{\text{prf-rand}} \\ \hline T := \text{empty} \\ \\ \underline{\textsc{query}(r):} \\ \quad \text{if } T[r] \text{ undefined:} \\ \quad\quad T[r] \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } T[r] \\ \hline \end{array}$$

Must use fact that r is unlikely to repeat (when chosen this way)

Security proof

CHALLENGE(m):
$r :=$ SAMP$()$
$z :=$ QUERY(r)
$x := z \oplus m$
return (r, x)

$\diamond$

$\mathcal{L}^F_{\text{prf-rand}}$
$T :=$ empty
QUERY(r):
if $T[r]$ undefined:
$T[r] \leftarrow \{0,1\}^{\text{out}}$
return $T[r]$

$\diamond$

$\mathcal{L}_{\text{samp-L}}$
SAMP$()$:
$r \leftarrow \{0,1\}^\lambda$
return r

Isolate sampling of r.

Security proof

CHALLENGE(m):
- $r := \textsc{samp}()$
- $z := \textsc{query}(r)$
- $x := z \oplus m$
- return (r, x)

$\diamond$

$\mathcal{L}^{F}_{\text{prf-rand}}$

$T :=$ empty

QUERY(r):
- if $T[r]$ undefined:
  - $T[r] \leftarrow \{0,1\}^{\text{out}}$
- return $T[r]$

$\diamond$

$\mathcal{L}_{\text{samp-L}}$

SAMP():
- $r \leftarrow \{0,1\}^{\lambda}$
- return r

Isolate sampling of r.

$\textsc{challenge}(m)$:
$r := \textsc{samp}()$
$z := \textsc{query}(r)$
$x := z \oplus m$
return (r, x)

$\diamond$

$\mathcal{L}^F_{\text{prf-rand}}$
$T :=$ empty
$\textsc{query}(r)$:
if $T[r]$ undefined:
$T[r] \leftarrow \{0,1\}^{\text{out}}$
return $T[r]$

$\diamond$

$\mathcal{L}_{\text{samp-R}}$
$R := \emptyset$
$\textsc{samp}()$:
$r \leftarrow \{0,1\}^\lambda \setminus R$
$R := R \cup \{r\}$
return r

Sample r without replacement (change $\mathcal{L}_{\text{samp-L}}$ to $\mathcal{L}_{\text{samp-R}}$).

$$\boxed{\begin{array}{l} \underline{\text{CHALLENGE}(m):} \\ \quad r := \text{SAMP}() \\ \quad z := \text{QUERY}(r) \\ \quad x := z \oplus m \\ \quad \text{return } (r, x) \end{array}} \diamond \boxed{\begin{array}{l} \quad\quad \mathcal{L}^{F}_{\text{prf-rand}} \\ \hline T := \text{empty} \\ \\ \underline{\text{QUERY}(r):} \\ \quad \text{if } T[r] \text{ undefined:} \\ \quad\quad T[r] \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } T[r] \end{array}} \diamond \boxed{\begin{array}{l} \quad\quad \mathcal{L}_{\text{samp-R}} \\ \hline R := \emptyset \\ \\ \underline{\text{SAMP}():} \\ \quad r \leftarrow \{0,1\}^{\lambda} \setminus R \\ \quad R := R \cup \{r\} \\ \quad \text{return } r \end{array}}$$

Sample r without replacement (change $\mathcal{L}_{\text{samp-L}}$ to $\mathcal{L}_{\text{samp-R}}$).

$$\begin{array}{|l|}\hline \underline{\text{CHALLENGE}(m):} \\ \quad r := \text{SAMP}() \\ \quad z := \text{QUERY}(r) \\ \quad x := z \oplus m \\ \quad \text{return } (r, x) \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}^{F}_{\text{prf-rand}} \\ \hline T := \text{empty} \\ \\ \underline{\text{QUERY}(r):} \\ \quad \text{if } T[r] \text{ undefined:} \\ \qquad T[r] \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } T[r] \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}_{\text{samp-R}} \\ \hline R := \emptyset \\ \\ \underline{\text{SAMP}():} \\ \quad r \leftarrow \{0,1\}^{\lambda} \setminus R \\ \quad R := R \cup \{r\} \\ \quad \text{return } r \\ \hline \end{array}$$

Now r values are **guaranteed** to never repeat.

Security proof

CHALLENGE(m):
$r :=$ SAMP()
$z :=$ QUERY(r)
$x := z \oplus m$
return (r, x)

$\diamond$

$\mathcal{L}^F_{\text{prf-rand}}$
$T :=$ empty
QUERY(r):
if $T[r]$ undefined:
$T[r] \leftarrow \{0,1\}^{\text{out}}$
return $T[r]$

$\diamond$

$\mathcal{L}_{\text{samp-R}}$
$R := \emptyset$
SAMP():
$r \leftarrow \{0,1\}^\lambda \setminus R$
$R := R \cup \{r\}$
return r

If-statement is always taken.

$$\boxed{\begin{array}{l} \underline{\textsc{challenge}(m):} \\ \quad r := \textsc{samp}() \\ \quad z := \textsc{query}(r) \\ \quad x := z \oplus m \\ \quad \text{return } (r, x) \end{array}} \diamond \boxed{\begin{array}{l} \underline{\textsc{query}(r):} \\ \quad z \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } z \end{array}} \diamond \boxed{\begin{array}{l} \mathcal{L}_{\text{samp-R}} \\ \hline R := \emptyset \\ \\ \underline{\textsc{samp}():} \\ \quad r \leftarrow \{0,1\}^{\lambda} \setminus R \\ \quad R := R \cup \{r\} \\ \quad \text{return } r \end{array}}$$

Middle library can therefore be simplified.

$$
\boxed{\begin{array}{l}
\underline{\textsc{challenge}(m):} \\
\quad r := \textsc{samp}() \\
\quad z := \textsc{query}(r) \\
\quad x := z \oplus m \\
\quad \text{return } (r, x)
\end{array}}
\diamond
\boxed{\begin{array}{l}
\underline{\textsc{query}(r):} \\
\quad z \leftarrow \{0,1\}^{\text{out}} \\
\quad \text{return } z
\end{array}}
\diamond
\boxed{\begin{array}{l}
\hfill \mathcal{L}_{\text{samp-R}} \hfill \\
\hline
R := \emptyset \\
\\
\underline{\textsc{samp}():} \\
\quad r \leftarrow \{0,1\}^{\lambda} \setminus R \\
\quad R := R \cup \{r\} \\
\quad \text{return } r
\end{array}}
$$

Middle library can therefore be simplified.

Security proof

CHALLENGE(m):
$r :=$ SAMP()
$z :=$ QUERY(r)
$x := z \oplus m$
return (r, x)

$\diamond$

QUERY(r):
$z \leftarrow \{0,1\}^{\text{out}}$
return z

$\diamond$

$\mathcal{L}_{\text{samp-R}}$
$R := \emptyset$
SAMP():
$r \leftarrow \{0,1\}^{\lambda} \setminus R$
$R := R \cup \{r\}$
return r

Inline call to QUERY.

$$\begin{array}{|l|}\hline \underline{\textsc{challenge}(m):} \\ \quad r := \textsc{samp}() \\ \quad z \leftarrow \{0,1\}^{\text{out}} \\ \quad x := z \oplus m \\ \quad \text{return } (r, x) \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}_{\text{samp-R}} \\ \hline R := \emptyset \\ \\ \underline{\textsc{samp}():} \\ \quad r \leftarrow \{0,1\}^{\lambda} \setminus R \\ \quad R := R \cup \{r\} \\ \quad \text{return } r \\ \hline \end{array}$$

Inline call to QUERY.

$$
\begin{array}{|l|}
\hline
\underline{\textsc{challenge}(m):} \\
\quad r := \textsc{samp}() \\
\quad z \leftarrow \{0,1\}^{\text{out}} \\
\quad x := z \oplus m \\
\quad \text{return } (r, x) \\
\hline
\end{array}
\diamond
\begin{array}{|l|}
\hline
\qquad \mathcal{L}_{\text{samp-R}} \\
\hline
R := \emptyset \\
\\
\underline{\textsc{samp}():} \\
\quad r \leftarrow \{0,1\}^{\lambda} \setminus R \\
\quad R := R \cup \{r\} \\
\quad \text{return } r \\
\hline
\end{array}
$$

Inline call to QUERY.

$$\begin{array}{|l|}\hline \underline{\textsc{challenge}(m):} \\ \quad r := \textsc{samp}() \\ \quad z \leftarrow \{0,1\}^{out} \\ \quad x := z \oplus m \\ \quad \text{return } (r, x) \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \qquad \mathcal{L}_{\text{samp-R}} \\ \hline R := \emptyset \\ \\ \underline{\textsc{samp}():} \\ \quad r \leftarrow \{0,1\}^{\lambda} \setminus R \\ \quad R := R \cup \{r\} \\ \quad \text{return } r \\ \hline \end{array}$$

Can apply the “one-time pad rule” (since mask z is uniform each time)

Security proof

$$\begin{array}{|l|}\hline \underline{\textsc{challenge}(m):} \\ \quad r := \textsc{samp}() \\ \quad x \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } (r, x) \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \qquad \mathcal{L}_{\text{samp-R}} \\ \hline R := \emptyset \\ \\ \underline{\textsc{samp}():} \\ \quad r \leftarrow \{0,1\}^{\lambda} \setminus R \\ \quad R := R \cup \{r\} \\ \quad \text{return } r \\ \hline \end{array}$$

Can apply the "one-time pad rule" (since mask z is uniform each time)

Security proof

$\underline{\textsc{challenge}(m)\text{:}}$ $r := \textsc{samp}()$ $x \leftarrow \{0,1\}^{\text{out}}$ return (r, x)	$\diamond$	$\mathcal{L}_{\text{samp-R}}$ $R := \emptyset$ $\underline{\textsc{samp}()\text{:}}$ $r \leftarrow \{0,1\}^\lambda \setminus R$ $R := R \cup \{r\}$ return r	

Can apply the "one-time pad rule" (since mask z is uniform each time)

Security proof

$$\begin{array}{|l|}\hline \underline{\textsc{challenge}(m):} \\ \quad r := \textsc{samp}() \\ \quad x \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } (r,x) \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}_{\text{samp-L}} \\ \hline \underline{\textsc{samp}():} \\ \quad r \leftarrow \{0,1\}^{\lambda} \\ \quad \text{return } r \\ \hline \end{array}$$

Replace $\mathcal{L}_{\text{samp-L}}$ with $\mathcal{L}_{\text{samp-R}}$.

$$\begin{array}{|l|}\hline \underline{\textsc{challenge}(m):} \\ \quad r := \textsc{samp}() \\ \quad x \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } (r, x) \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \quad \mathcal{L}_{\text{samp-L}} \\ \hline \underline{\textsc{samp}():} \\ \quad r \leftarrow \{0,1\}^{\lambda} \\ \quad \text{return } r \\ \hline \end{array}$$

Replace $\mathcal{L}_{\text{samp-L}}$ with $\mathcal{L}_{\text{samp-R}}$.

$$\begin{array}{|l|}\hline \underline{\textsc{challenge}(m):} \\ \quad r := \textsc{samp}() \\ \quad x \leftarrow \{0,1\}^{\text{out}} \\ \quad \text{return } (r,x) \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}_{\text{samp-L}} \\ \hline \underline{\textsc{samp}():} \\ \quad r \leftarrow \{0,1\}^{\lambda} \\ \quad \text{return } r \\ \hline \end{array}$$

Inline call to SAMP.

$\textsc{challenge}(m)$:

$r \leftarrow \{0,1\}^{\lambda}$
$x \leftarrow \{0,1\}^{\text{out}}$
return (r, x)

Inline call to SAMP.

$$\boxed{\begin{array}{l}\underline{\textsc{challenge}(m):}\\ \quad r \leftarrow \{0,1\}^{\lambda}\\ \quad x \leftarrow \{0,1\}^{\text{out}}\\ \quad \text{return } (r, x)\end{array}}$$

Inline call to SAMP.

$\mathcal{L}^{\Sigma}_{\text{cpa\$-rand}}$
$\underline{\textsc{challenge}(m):}$ $r \leftarrow \{0,1\}^\lambda$ $x \leftarrow \{0,1\}^{\text{out}}$ return (r, x)

But every response is chosen uniformly: This is just $\mathcal{L}_{\text{cpa\$-rand}}$.