One-time pad security:

OTP:

$$\mathcal{K} = \{0,1\}^\lambda \quad \mathcal{M} = \{0,1\}^\lambda \quad \mathcal{C} = \{0,1\}^\lambda$$

```
KeyGen:            Enc(k, m):           Dec(k, c):
  k ← 𝒦               return k ⊕ m         return k ⊕ c
  return k
```

Claim:

OTP satisfies one-time secrecy. That is, $\mathcal{L}^{\text{OTP}}_{\text{ots-L}} \equiv \mathcal{L}^{\text{OTP}}_{\text{ots-R}}$.

We will **use** the fact that OTP ciphertexts are uniformly distributed:

```
ctxt(m ∈ {0,1}^λ):          ctxt(m ∈ {0,1}^λ):
  k ← {0,1}^λ          ≡      c ← {0,1}^λ
  return k ⊕ m                return c
```

Overview:

Want to show:

$\mathcal{L}^{\text{OTP}}_{\text{ots-L}}$		$\mathcal{L}^{\text{OTP}}_{\text{ots-R}}$
$\textsc{query}(m_L, m_R \in \text{OTP}.\mathcal{M})$: $k \leftarrow \text{OTP.KeyGen}$ $c := \text{OTP.Enc}(k, m_L)$ return c	$\equiv$	$\textsc{query}(m_L, m_R \in \text{OTP}.\mathcal{M})$: $k \leftarrow \text{OTP.KeyGen}$ $c := \text{OTP.Enc}(k, m_R)$ return c

Standard hybrid technique:

- Starting with $\mathcal{L}^{\text{OTP}}_{\text{ots-L}}$, make a sequence of small modifications
- Each modification has no effect on calling program / adversary
- Sequence of modifications ends with $\mathcal{L}^{\text{OTP}}_{\text{ots-R}}$

Security proof

$\mathcal{L}_{\text{ots-L}}^{\text{OTP}}$

$\textsc{query}(m_L, m_R \in \text{OTP}.\mathcal{M})$:
$k \leftarrow \text{OTP.KeyGen}$
$c := \text{OTP.Enc}(k, m_L)$
return c

Starting point is $\mathcal{L}_{\text{ots-L}}^{\text{OTP}}$.

Security proof

$\mathcal{L}_{\text{ots-L}}^{\text{OTP}}$

$\underline{\text{QUERY}(m_L, m_R \in \text{OTP}.\mathcal{M}):}$
  $k \leftarrow \text{OTP.KeyGen}$
  $c := \text{OTP.Enc}(k, m_L)$
  return c

Starting point is $\mathcal{L}_{\text{ots-L}}^{\text{OTP}}$. Fill in details of OTP

Security proof

$\mathcal{L}^{\text{OTP}}_{\text{ots-L}}$

$\underline{\text{QUERY}(m_L, m_R \in \{0,1\}^\lambda):}$

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

$c := k \oplus m_L$

return c

Starting point is $\mathcal{L}^{\text{OTP}}_{\text{ots-L}}$. Fill in details of OTP

Security proof

$\mathcal{L}_{\text{ots-L}}^{\text{OTP}}$
$\text{QUERY}(m_L, m_R \in \{0,1\}^\lambda)$:
$k \leftarrow \{0,1\}^\lambda$
$c := k \oplus m_L$
return c

Starting point is $\mathcal{L}_{\text{ots-L}}^{\text{OTP}}$. Fill in details of OTP

$\mathcal{L}^{\text{OTP}}_{\text{ots-L}}$

```
QUERY(m_L, m_R ∈ {0,1}^λ):
  k ← {0,1}^λ
  c := k ⊕ m_L
  return c
```

These statements appear also in $\mathcal{L}_{\text{otp-real}}$.

$\text{QUERY}(m_L, m_R \in \{0,1\}^\lambda)$:
$c := \text{CTXT}(m_L)$
return c

$\diamond$

$\mathcal{L}_{\text{otp-real}}$
$\text{CTXT}(m \in \{0,1\}^\lambda)$:
$k \leftarrow \{0,1\}^\lambda$
return $k \oplus m$

Factor out so that $\mathcal{L}_{\text{otp-real}}$ appears.

$$\boxed{\begin{array}{l}\underline{\textsc{query}(m_L, m_R \in \{0,1\}^\lambda):}\\ \quad c := \textsc{ctxt}(m_L)\\ \quad \text{return } c\end{array}} \diamond \boxed{\begin{array}{c}\mathcal{L}_{\text{otp-real}}\\ \hline \begin{array}{l}\underline{\textsc{ctxt}(m \in \{0,1\}^\lambda):}\\ \quad k \leftarrow \{0,1\}^\lambda\\ \quad \text{return } k \oplus m\end{array}\end{array}}$$

Factor out so that $\mathcal{L}_{\text{otp-real}}$ appears.

Security proof

$$\boxed{\begin{array}{l}\underline{\textsc{query}(m_L, m_R \in \{0,1\}^\lambda):}\\ \quad c := \textsc{ctxt}(m_L)\\ \quad \text{return } c\end{array}} \diamond \boxed{\begin{array}{c}\mathcal{L}_{\text{otp-rand}}\\ \hline \begin{array}{l}\underline{\textsc{ctxt}(m \in \{0,1\}^\lambda):}\\ \quad c \leftarrow \{0,1\}^\lambda\\ \quad \text{return } c\end{array}\end{array}}$$

$\mathcal{L}_{\text{otp-real}}$ can be replaced with $\mathcal{L}_{\text{otp-rand}}$.

$\textsc{query}(m_L, m_R \in \{0,1\}^\lambda)$:
$c := \textsc{ctxt}(m_L)$
return c

$\diamond$

$\mathcal{L}_{\text{otp-rand}}$
$\textsc{ctxt}(m \in \{0,1\}^\lambda)$:
$c \leftarrow \{0,1\}^\lambda$
return c

$\mathcal{L}_{\text{otp-real}}$ can be replaced with $\mathcal{L}_{\text{otp-rand}}$.

Security proof

QUERY($m_L, m_R \in \{0,1\}^\lambda$):
$c :=$ CTXT(m_L)
return c

$\diamond$

$\mathcal{L}_{\text{otp-rand}}$
CTXT($m \in \{0,1\}^\lambda$):
$c \leftarrow \{0,1\}^\lambda$
return c

Argument to CTXT is never used!

$\text{QUERY}(m_L, m_R \in \{0,1\}^\lambda)$:
$c := \text{CTXT}(m_R)$
return c

$\diamond$

$\mathcal{L}_{\text{otp-rand}}$
$\text{CTXT}(m \in \{0,1\}^\lambda)$:
$c \leftarrow \{0,1\}^\lambda$
return c

Unused argument can be changed to m_R.

Security proof

$$\begin{array}{|l|}\hline \underline{\textsc{query}(m_L, m_R \in \{0,1\}^\lambda):} \\ \quad c := \textsc{ctxt}(m_R) \\ \quad \text{return } c \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}_{\text{otp-rand}} \\ \hline \underline{\textsc{ctxt}(m \in \{0,1\}^\lambda):} \\ \quad c \leftarrow \{0,1\}^\lambda \\ \quad \text{return } c \\ \hline \end{array}$$

Unused argument can be changed to m_R.

$$\begin{array}{|l|}\hline \underline{\textsc{query}(m_L, m_R \in \{0,1\}^\lambda):} \\ \quad c := \textsc{ctxt}(m_R) \\ \quad \text{return } c \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}_{\text{otp-real}} \\ \hline \underline{\textsc{ctxt}(m \in \{0,1\}^\lambda):} \\ \quad k \leftarrow \{0,1\}^\lambda \\ \quad \text{return } k \oplus m \\ \hline \end{array}$$

$\mathcal{L}_{\text{otp-rand}}$ can be replaced with $\mathcal{L}_{\text{otp-real}}$.

$$\begin{array}{|l|}\hline \underline{\textsc{query}(m_L, m_R \in \{0,1\}^\lambda):} \\ \quad c := \textsc{ctxt}(m_R) \\ \quad \text{return } c \\ \hline \end{array} \diamond \begin{array}{|l|}\hline \mathcal{L}_{\text{otp-real}} \\ \hline \underline{\textsc{ctxt}(m \in \{0,1\}^\lambda):} \\ \quad k \leftarrow \{0,1\}^\lambda \\ \quad \text{return } k \oplus m \\ \hline \end{array}$$

$\mathcal{L}_{\text{otp-rand}}$ can be replaced with $\mathcal{L}_{\text{otp-real}}$.

$\textsc{query}(m_L, m_R \in \{0,1\}^\lambda)$:
$c := \textsc{ctxt}(m_R)$
return c

$\diamond$

$\mathcal{L}_{\text{otp-real}}$
$\textsc{ctxt}(m \in \{0,1\}^\lambda)$:
$k \leftarrow \{0,1\}^\lambda$
return $k \oplus m$

Inline the subroutine call.

$\textsc{query}(m_L, m_R \in \{0,1\}^\lambda)$:
$k \leftarrow \{0,1\}^\lambda$
$c := k \oplus m_R$
return c

Inline the subroutine call.

$$\begin{array}{l}\underline{\textsc{query}(m_L, m_R \in \{0,1\}^\lambda):} \\ \quad k \leftarrow \{0,1\}^\lambda \\ \quad c := k \oplus m_R \\ \quad \text{return } c\end{array}$$

Inline the subroutine call.

$$\begin{array}{|l|}\hline \textsc{query}(m_L, m_R \in \{0,1\}^\lambda): \\ \hline \quad k \leftarrow \{0,1\}^\lambda \\ \quad c := k \oplus m_R \\ \quad \text{return } c \\ \hline \end{array}$$

This happens to be $\mathcal{L}^{\text{OTP}}_{\text{ots-R}}$.

```
QUERY(m_L, m_R ∈ {0,1}^λ):
  k ← {0,1}^λ
  c := k ⊕ m_R
  return c
```

This happens to be $\mathcal{L}^{\text{OTP}}_{\text{ots-R}}$.

$\mathcal{L}^{\text{OTP}}_{\text{ots-R}}$

$\underline{\text{QUERY}(m_L, m_R \in \text{OTP}.\mathcal{M}):}$
$\quad k \leftarrow \text{OTP.KeyGen}$
$\quad c := \text{OTP.Enc}(k, m_R)$
$\quad \text{return } c$

This happens to be $\mathcal{L}^{\text{OTP}}_{\text{ots-R}}$.

$\mathcal{L}^{\text{OTP}}_{\text{ots-R}}$

```
QUERY(m_L, m_R ∈ OTP.M):
  k ← OTP.KeyGen
  c := OTP.Enc(k, m_R)
  return c
```

This happens to be $\mathcal{L}^{\text{OTP}}_{\text{ots-R}}$.

Summary

We showed:

$\mathcal{L}^{\text{OTP}}_{\text{ots-L}}$

$\textsc{query}(m_L, m_R \in \text{OTP}.\mathcal{M})$:

$k \leftarrow \text{OTP.KeyGen}$
$c := \text{OTP.Enc}(k, m_L)$
return c

$\equiv \cdots \equiv$

$\mathcal{L}^{\text{OTP}}_{\text{ots-R}}$

$\textsc{query}(m_L, m_R \in \text{OTP}.\mathcal{M})$:

$k \leftarrow \text{OTP.KeyGen}$
$c := \text{OTP.Enc}(k, m_R)$
return c

So OTP satisfies one-time secrecy.