CS-E4340 Cryptography: Exercise Sheet 5

—Symmetric Encryption Schemes (SE)—

Submission Deadline: October 10, 11:30 via MyCourses

Each exercise can give up to two participation points, 2 for a mostly correct solution and 1 point for a good attempt. Overall, the exercise sheet gives at most 4 participation points.

Exercise Sheet 5 is intended to help...

(a) ...understanding the security games indistinguishability under chosen plaintext attacks (IND-CPA) and the authenticated encryption security (AE) for a symmetric encryption scheme (SE).
(b) ...familiarizing yourself further with the notion of a reduction and learning to carry out reductions yourself. In particular, you will be able to practice inlining and spotting errors in an inlining proof.

Exercise 1 then follows the tradition and shows that some schemes that look somewhat unnatural can still be secure.
Exercise 2 returns to the question of what is a good definition.
Exercise 3 considers a whole class of encryption schemes at the same time and shows that they can never be secure according to our definition.
Exercise 4 takes up the challenge from lecture 5 on how to combine MACs and IND-CPA-secure encryption schemes into an authenticated encryption scheme.
Exercise 5 considers a quite canonical encryption scheme.
Ex. 2, 3 & 4 ask for adversary constructions.
Ex. 1, 4 & 5 cover security proofs via reduction.

Solution 1. The scheme is not efficient and it is also not (guaranteed to be) correct. To elaborate:

Efficiency: Consider the line **find** r **such that** $r' = f(k, r)$. This function *might* be efficient if f is a PRP, but currently, we are only guaranteed that f is a PRF, so inversion is not necessarily efficient. In general, we do not know better than do exhaustive search for solving the problem of finding an inverse.

If we had been given the task to prove that inversion is hard, we could have argued via the following counterexample: consider the following PRF f, constructed from a PRF f_{prp} and a one-way permutation f_{owp}. Define $f(k, x) := f_{\text{prp}}(k, f_{\text{owp}}(x))$. Now, if we can invert f, then we can invert $f_{\text{owp}}(x)$ and obtain a contradiction.

Correctness: A secure PRF can have an image space that is a small (even negl.) part of the $\{0,1\}^\lambda$. Consequently the r found in decryption is not necessarily the correct one (it might be correct with only negl. probability!)

Exercise 1. (Candidate Encryption Schemes) Let se be an IND-CPA secure encryption scheme.

1. Do you think these encryption schemes are IND-CPA secure? Justify your intuition for each scheme.

2. Choose one of the secure schemes and prove security by giving a reduction (in pseudocode). In this exercise, it suffices to give an intuitive explanation for why the adversary works.

$se_1.enc(k, m)$
$c' \leftarrow\$ se.enc(k, m)$
$c \leftarrow c'||1$
return c

$se_1.dec(k, c)$
$c' \leftarrow c[1..|c| - 1]$
$m \leftarrow se.dec(k, c')$
return m

$se_2.enc(k, m)$
$m' \leftarrow m||0$
$c \leftarrow\$ se.enc(k, m')$
return c

$se_2.dec(k, c)$
$m' \leftarrow se.dec(k, c)$
$m \leftarrow m'[1..|m| - 1]$
return m

$se_3.enc(k, m)$
$c_0 \leftarrow\$ se.enc(k, m)$
$c_1 \leftarrow\$ se.enc(k, m)$
$c \leftarrow (c_0, c_1)$
return c

$se_3.dec(k, c)$
parse $(c_0, c_1) \leftarrow c$
$m \leftarrow se.dec(k, c_0)$
$m' \leftarrow se.dec(k, c_1)$
if $m = m'$ **return** m
else return $\perp$

Solution 2. All the schemes are actually IND-CPA secure. First observe that the schemes are efficient if se is efficient, since they only append or remove single bits or run the algorithms of se twice.
For correctness:

$$\begin{aligned}
&\Pr[se_1.dec(k, se_1.enc(k, m)) = m] \\
=&\Pr[se_1.dec(k, se.enc(k, m)\|1) = m] \\
=&\Pr[se.dec(k, (se.enc(k, m)\|1)[1, \ldots, |c| - 1]) = m] \\
=&\Pr[se.dec(k, se.enc(k, m)) = m] \\
=&1,
\end{aligned}$$

where the latter equality is due to the correctness of se.

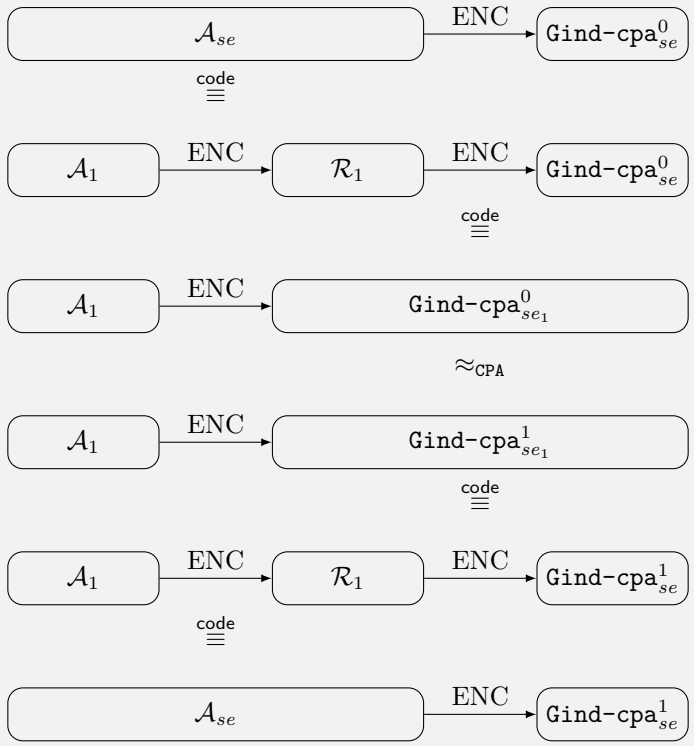
$$\begin{aligned}
&\Pr[se_2.dec(k, se_2.enc(k, m)) = m] \\
=&\Pr[se_2.dec(k, se.enc(k, m\|0))] = m \\
=&\Pr[m'[1, \ldots, |m'| - 1] = m \mid m' \leftarrow\$ se.dec(k, se.enc(k, m\|0))] \\
=&\Pr[m'[1, \ldots, |m'| - 1] = m \mid m' \leftarrow m||0] \\
=&\Pr[m'[1, \ldots, m] = m \mid m' \leftarrow m||0] \\
=&\Pr[m[1, \ldots, m] = m] \\
=&1,
\end{aligned}$$

where the third equality is due to the correctness of se.

$$\begin{aligned}
&\Pr[se_2.dec(k, se_2.enc(k, m)) = m] \\
=&\Pr_{c_0 \leftarrow\$ se.enc(k,m), c_1 \leftarrow\$ se.enc(k,m)}[se_2.dec(k, (c_0, c_1)) = m] \\
=&\Pr_{c_0 \leftarrow\$ se.enc(k,m), c_1 \leftarrow\$ se.enc(k,m)}[se.dec(k, c_0) = se.dec(k, c_1)] \\
=&\Pr_{c_0 \leftarrow\$ se.enc(k,m), c_1 \leftarrow\$ se.enc(k,m)}[m = m] \\
=&1,
\end{aligned}$$

where the second equality is due to the correctness of se.

Let $\mathcal{A}_1, \mathcal{A}_2$, and $\mathcal{A}_3$ be the adversaries against se_1, se_2, and se_3, respectively. We construct reductions $\mathcal{R}^{\mathcal{A}_1}, \mathcal{R}^{\mathcal{A}_2}$, and $\mathcal{R}^{\mathcal{A}_3}$. These will result in adversaries (for example, $\mathcal{A}_1 \to \mathcal{R}^{\mathcal{A}_1}$ against se which we will call $\mathcal{A}_{se}$). The proofs follow the following (graphical) outlay:



Security of se_1: First define the reduction as

$\underline{\underline{\mathcal{R}^{\mathcal{A}_1}}}$
$\underline{\mathsf{ENC}(x)}$
$c' \leftarrow \mathsf{ENC}(x)$
$c \leftarrow c' \| 1$
return c

We now show that $\mathcal{R}^{\mathcal{A}_1} \to \texttt{Gind-cpa}^b_{se} = \texttt{Gind-cpa}^b_{se_1}$ for $b \in \{0, 1\}$

$\underline{\underline{\mathcal{R}_1 \to \texttt{Gind-cpa}^0_{se}}}$
$\underline{\mathsf{ENC}(x)}$

// Inline Starts Here
if $k = \bot$:
  $k \leftarrow\$ \{0,1\}^\lambda$

$c' \leftarrow se.enc(k, x)$
// Inline Ends Here
$c \leftarrow c' \| 1$

return c

$\underline{\underline{\mathcal{R}_1 \to \texttt{Gind-cpa}^0_{se}}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
  $k \leftarrow\$ \{0,1\}^\lambda$
// se_1 To Be Outlined
$c' \leftarrow se.enc(k, x)$

$c \leftarrow c' \| 1$
// Outline Ends Here
return c

$\underline{\underline{\texttt{Gind-cpa}^0_{se_1}}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
  $k \leftarrow\$ \{0,1\}^\lambda$

$c \leftarrow se_1.enc(k, x)$

return c

$\underline{\underline{\mathcal{R}_1 \to \texttt{Gind-cpa}^1_{se}}}$
$\underline{\mathsf{ENC}(x)}$

// Inline Starts Here
if $k = \bot$:
  $k \leftarrow\$ \{0,1\}^\lambda$
$x' \leftarrow 0^{|x|}$

$c' \leftarrow se.enc(k, x')$
// Inline Ends Here
$c \leftarrow c' \| 1$

return c

$\underline{\underline{\mathcal{R}_1 \to \texttt{Gind-cpa}^1_{se}}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
  $k \leftarrow\$ \{0,1\}^\lambda$
$x' \leftarrow 0^{|x|}$
// se_1 To Be Outlined
$c' \leftarrow se.enc(k, x')$

$c \leftarrow c' \| 1$
// Outline Ends Here
return c

$\underline{\underline{\texttt{Gind-cpa}^1_{se_1}}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
  $k \leftarrow\$ \{0,1\}^\lambda$
$x' \leftarrow 0^{|x|}$

$c \leftarrow se_1.enc(k, x')$

return c

Therefore

$$\begin{aligned}\mathbf{Adv}(\mathcal{A}_1; \texttt{ind-cpa}^0_{se_1}, \texttt{Gind-cpa}^1_{se_1}) &= \mathbf{Adv}(\mathcal{A}_1; \mathcal{R}_1 \to \texttt{Gind-cpa}^0_{se}, \mathcal{R}_1 \to \texttt{Gind-cpa}^1_{se}) \\ &= \mathbf{Adv}(\mathcal{A}_1 \to \mathcal{R}_1; \texttt{Gind-cpa}^0_{se}, \texttt{Gind-cpa}^1_{se})\end{aligned}$$

and we have constructed an adversary $\mathcal{A}_1 \to \mathcal{R}_1$ against IND-CPA security of se (with the same advantage) if $\mathcal{A}_1$ is an adversary against se_1. □

Security of se_2: First define the reduction as

$\underline{\underline{\mathcal{R}_2}}$
$\underline{\mathsf{ENC}(x)}$
$x' \leftarrow x \| 0$
$c \leftarrow \mathsf{ENC}(x)$
return c

We now show that $\mathcal{R}_2 \to \texttt{Gind-cpa}^b_{se} = \texttt{Gind-cpa}^b_{se_2}$ for $b \in \{0,1\}$

$\underline{\mathcal{R}_2 \to \texttt{Gind-cpa}^0_{se}}$
$\underline{\mathsf{ENC}(x)}$

$x' \leftarrow x\|0$
// Inline Starts Here
if $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$
$c \leftarrow se.enc(k, x')$
// Inline Ends Here
return c

$\underline{\mathcal{R}_2 \to \texttt{Gind-cpa}^0_{se}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$
$x' \leftarrow x\|0$
$c \leftarrow se.enc(k, x')$
return c

$\underline{\mathcal{R}_2 \to \texttt{Gind-cpa}^0_{se}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$
// se_2 To Be Outlined
$x' \leftarrow x\|0$
$c \leftarrow se.enc(k, x')$
// Outline Ends Here
return c

$\underline{\texttt{Gind-cpa}^0_{se_2}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$
$c \leftarrow se.enc_2(k, x)$
return c

$\underline{\mathcal{R}_2 \to \texttt{Gind-cpa}^1_{se}}$
$\underline{\mathsf{ENC}(x)}$

$x' \leftarrow x\|0$
// Inline Starts Here
if $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$
$x'' \leftarrow 0^{|x'|}$
$c \leftarrow se.enc(k, x'')$
// Inline Ends Here
return c

$\underline{\mathcal{R}_2 \to \texttt{Gind-cpa}^1_{se}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$
$x' \leftarrow 0^{|x|}$
$x'' \leftarrow x'\|0$
$c \leftarrow se.enc(k, x'')$
return c

$\underline{\mathcal{R}_2 \to \texttt{Gind-cpa}^1_{se}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$
$x' \leftarrow 0^{|x|}$
// se_2 To Be Outlined
$x'' \leftarrow x'\|0$
$c \leftarrow se.enc(k, x'')$
// Outline Ends Here
return c

$\underline{\texttt{Gind-cpa}^1_{se_2}}$
$\underline{\mathsf{ENC}(x)}$

if $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$
$x' \leftarrow 0^{|x|}$
$c \leftarrow se.enc_2(k, x')$
return c

Note there is one slightly non-trivial step in this transformation. We need to use that $0^{|x|}\|0 = 0^{|x\|0|}$.
Therefore

$$\begin{aligned}\mathbf{Adv}(\mathcal{A}_2; \texttt{ind-cpa}^0_{se_2}, \texttt{Gind-cpa}^1_{se_2}) &= \mathbf{Adv}(\mathcal{A}_2; \mathcal{R}_2 \to \texttt{Gind-cpa}^0_{se}, \mathcal{R}_2 \to \texttt{Gind-cpa}^1_{se}) \\ &= \mathbf{Adv}(\mathcal{A}_2 \to \mathcal{R}_2; \texttt{Gind-cpa}^0_{se}, \texttt{Gind-cpa}^1_{se})\end{aligned}$$

and we have constructed an adversary $\mathcal{A}_2 \to \mathcal{R}_2$ against IND-CPA security of se (with the same advantage) if $\mathcal{A}_2$ is an adversary against se_2. □

Security of se_3:

$\underline{\mathcal{R}_3}$
$\underline{\mathsf{ENC}(x)}$

$c_1 \leftarrow \mathsf{ENC}(x)$
$c_2 \leftarrow \mathsf{ENC}(x)$
return (c_1, c_2)

We now show that $\mathcal{R}_3 \to \texttt{Gind-cpa}^b_{se} = \texttt{Gind-cpa}^b_{se_3}$ for $b \in \{0,1\}$

$\mathcal{R}_3 \to \texttt{Gind-cpa}^0_{se}$	$\mathcal{R}_3 \to \texttt{Gind-cpa}^0_{se}$	$\mathcal{R}_3 \to \texttt{Gind-cpa}^0_{se}$	$\texttt{Gind-cpa}^0_{se_3}$
$\mathsf{ENC}(x)$	$\mathsf{ENC}(x)$	$\mathsf{ENC}(x)$	$\mathsf{ENC}(x)$
// Inline Starts Here			
if $k = \bot$:	**if** $k = \bot$:	**if** $k = \bot$:	**if** $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$	$k \leftarrow\$ \{0,1\}^\lambda$	$k \leftarrow\$ \{0,1\}^\lambda$	$k \leftarrow\$ \{0,1\}^\lambda$
		// se_3 To Be Outlined	
$c_1 \leftarrow se.enc(k, x)$	$c_1 \leftarrow se.enc(k, x)$	$c_1 \leftarrow se.enc(k, x)$	
// Inline Ends Here			
// Inline Starts Here			$c \leftarrow se_3.enc(k, x)$
if $k = \bot$:			
$k \leftarrow\$ \{0,1\}^\lambda$			
$c_2 \leftarrow se.enc(k, x)$	$c_2 \leftarrow se.enc(k, x)$	$c_2 \leftarrow se.enc(k, x)$	
// Inline Ends Here		// Outline Ends Here	
return (c_1, c_2)	**return** (c_1, c_2)	**return** (c_1, c_2)	**return** c

$\mathcal{R}_3 \to \texttt{Gind-cpa}^1_{se}$	$\mathcal{R}_3 \to \texttt{Gind-cpa}^1_{se}$	$\mathcal{R}_3 \to \texttt{Gind-cpa}^1_{se}$	$\texttt{Gind-cpa}^1_{se_3}$
$\mathsf{ENC}(x)$	$\mathsf{ENC}(x)$	$\mathsf{ENC}(x)$	$\mathsf{ENC}(x)$
// Inline Starts Here			
if $k = \bot$:	**if** $k = \bot$:	**if** $k = \bot$:	**if** $k = \bot$:
$k \leftarrow\$ \{0,1\}^\lambda$	$k \leftarrow\$ \{0,1\}^\lambda$	$k \leftarrow\$ \{0,1\}^\lambda$	$k \leftarrow\$ \{0,1\}^\lambda$
$x' \leftarrow 0^{\lvert x \rvert}$	$x' \leftarrow 0^{\lvert x \rvert}$	$x' \leftarrow 0^{\lvert x \rvert}$	$x' \leftarrow 0^{\lvert x \rvert}$
		// se_3 To Be Outlined	
$c_1 \leftarrow se.enc(k, x')$	$c_1 \leftarrow se.enc(k, x')$	$c_1 \leftarrow se.enc(k, x')$	
// Inline Ends Here			
// Inline Starts Here			
if $k = \bot$:			$c \leftarrow se_3.enc(k, x')$
$k \leftarrow\$ \{0,1\}^\lambda$			
$x'' \leftarrow 0^{\lvert x \rvert}$			
$c_2 \leftarrow se.enc(k, x'')$	$c_2 \leftarrow se.enc(k, x')$	$c_2 \leftarrow se.enc(k, x')$	
// Inline Ends Here		// Outline Ends Here	
return (c_1, c_2)	**return** (c_1, c_2)	**return** (c_1, c_2)	**return** c

Therefore

$$\begin{aligned}\mathbf{Adv}(\mathcal{A}_3; \texttt{ind-cpa}^0_{se_2}, \texttt{Gind-cpa}^1_{se_2}) &= \mathbf{Adv}(\mathcal{A}_3; \mathcal{R}_3 \to \texttt{Gind-cpa}^0_{se}, \mathcal{R}_3 \to \texttt{Gind-cpa}^1_{se}) \\ &= \mathbf{Adv}(\mathcal{A}_3 \to \mathcal{R}_3; \texttt{Gind-cpa}^0_{se}, \texttt{Gind-cpa}^1_{se})\end{aligned}$$

and we have constructed an adversary $\mathcal{A}_3 \to \mathcal{R}_3$ against IND-CPA security of se (with the same advantage) if $\mathcal{A}_3$ is an adversary against se_3. □

Note that our constructed adversary against se will make two oracle queries for each oracle query of $\mathcal{A}_3$.

Exercise 2. (Different Definitions) The goal of AE security is to make sure an adversary is unable to "come up" with a decrypting ciphertext on its own. Consider the following approach to capture this property. Let the games $\texttt{Gae'}^0_{se}$ and $\texttt{Gae'}^1_{se}$ be defined as

$\texttt{Gae'}^0_{se}$

Parameters

λ:	sec. parameter
se:	sym. enc. sch.

Package State

k:	key

ENC(x)

if $k = \bot$:
  $k \leftarrow\$ \{0,1\}^\lambda$
$c \leftarrow\$ se.enc(k, x)$
return c

DEC(c)

if $k = \bot$:
  $k \leftarrow\$ \{0,1\}^\lambda$
$x \leftarrow se.dec(k, c)$
return x

$\texttt{Gae'}^1_{se}$

Parameters

λ:	sec. parameter
se:	sym. enc. sch.

Package State

k:	key
$\mathcal{L}$:	set

ENC(x)

if $k = \bot$:
  $k \leftarrow\$ \{0,1\}^\lambda$
$x' \leftarrow 0^{|x|}$
$c \leftarrow\$ se.enc(k, x')$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{c\}$
return c

DEC(c)

if $c \in \mathcal{L}$
  $x \leftarrow se.dec(k, c)$
  return x
else
  return $\bot$

Show that no correct encryption scheme is secure according to this definition by giving a successful attacker against the AE' security (in pseudocode) and analyze its success probability.

Solution 3. The problem with this definition comes from the fact that in the ideal game, instead of the message, a zero-message is encrypted. In the proper definition this does not matter as the ideal decryption just looks in the table and returns the original message. This definition tries to be smart and calls decryption in any case, therefore in the ideal world it will always decrypt to $0^{|x|}$

$\mathcal{A}$

$m \leftarrow 1$
$c \leftarrow$ ENC(m)
$m' \leftarrow$ DEC(c)
if $m' = 0$
  return 1
else
  return 0

Claim:

$$\mathbf{Adv}(\mathcal{A}; \texttt{Gae'}^0_{se}, \texttt{Gae'}^1_{se}) = \big|\Pr\big[\mathcal{A} \to \texttt{Gae'}^0_{se} = 1\big] - \Pr\big[\mathcal{A} \to \texttt{Gae'}^1_{se} = 1\big]\big|$$

is non-negligible.

Success Probability

$$\begin{aligned}
\mathbf{Adv}(\mathcal{A}; Gae'^0, Gae'^1) &= \big|\Pr\big[\mathcal{A} \to \texttt{Gae'}^0_{se} = 1\big] - \Pr\big[\mathcal{A} \to \texttt{Gae'}^1_{se} = 1\big]\big| \\
&= \big|0 - \Pr\big[\mathcal{A} \to \texttt{Gae'}^1_{se} = 1\big]\big| \\
&= \big|0 - \Pr\big[c \leftarrow \mathsf{ENC}^1(1); \mathsf{DEC}^1(c) = 0\big]\big| \\
&= \big|0 - \Pr_{k \leftarrow\$ \{0,1\}^\lambda}[c \leftarrow se.enc(k, 0); se.dec(k, c) = 0]\big| \\
&= \big|0 - 1\big| = 1
\end{aligned}$$

The first equality holds due to the correctness of the encryption scheme. The second and third by inlining the adversary (2) and the game(3) and equality 4 finally by correctness again.

$\mathcal{A}$	$\mathcal{A} \to \texttt{Gae'}^1$	$\mathcal{A} \to \texttt{Gae'}^1$
$m \leftarrow 1$	$m \leftarrow 1$	$m \leftarrow 1$
$c \leftarrow \mathsf{ENC}(m)$	// ENC Oracle	// ENC Oracle
$m' \leftarrow \mathsf{DEC}(c)$	**if** $k = \perp$:	
if $m' = 0$	$k \leftarrow\$ \{0,1\}^\lambda$	$k \leftarrow\$ \{0,1\}^\lambda$
return 1	$x' \leftarrow 0^{\lvert m \rvert}$	$x' \leftarrow 0$
else	$c \leftarrow\$ se.enc(k, x')$	$c \leftarrow\$ se.enc(k, x')$
return 0	$\mathcal{L} \leftarrow \mathcal{L} \cup \{c\}$	
	// DEC Oracle	// DEC Oracle
	if $c \in \mathcal{L}$	
	$m \leftarrow dec(k, c)$	$m \leftarrow dec(k, c)$
	else	
	$m \leftarrow \perp$	
	// End Oracle Calls	// End Oracle Calls
	if $m' = 0$	**if** $m' = 0$
	return 1	**return** 1
	else	**else**
	return 0	**return** 0

Claim:

$$\mathbf{Adv}(\mathcal{A}; \texttt{Gae'}^0_{se}, \texttt{Gae'}^0_{se}) = \big|\Pr\big[\mathcal{A} \to \texttt{Gae'}^0_{se} = 1\big] - \Pr\big[\mathcal{A} \to \texttt{Gae'}^1_{se} = 1\big]\big|$$

is non-negligible.

Success Probability

$$\begin{aligned}
\mathbf{Adv}(\mathcal{A}; Gae'^0, Gae'^1) &= \big|\Pr\big[\mathcal{A} \to \texttt{Gae'}^0_{se} = 1\big] - \Pr\big[\mathcal{A} \to \texttt{Gae'}^1_{se} = 1\big]\big| \\
&= \big|0 - \Pr\big[\mathcal{A} \to \texttt{Gae'}^1_{se} = 1\big]\big| \\
&= \big|0 - \Pr_{k \leftarrow\$ \{0,1\}^\lambda}[c \leftarrow se.enc(k, 0); se.dec(k, c) = 0]\big| \\
&= \big|0 - 1\big| = 1
\end{aligned}$$

Which is clearly non-negl. The first equality holds due to the correctness of the encryption scheme. The second by inlining $\mathcal{A} \to \texttt{Gae'}^1$ and equality 3 finally by correctness again.

Exercise 3. (Deterministic Encryption Schemes[1]) Consider a different encryption scheme $se' = (se'.kgen, se'.enc, se'.dec)$ such that $se'.enc$ is *deterministic*. Show that se' is not IND-CPA-secure by giving a successful attacker against the IND-CPA security of se' (in pseudocode) and analyze its success probability.

$\mathcal{A}$

$r \leftarrow 0$
$r' \leftarrow 1$
$c \leftarrow \mathsf{ENC}(r)$
$c' \leftarrow \mathsf{ENC}(r')$
if $c = c'$
  return 1
else
  return 0

Consider an adversary that submits two unique messages. In the ideal world $\texttt{Gind-cpa}^1_{se}$, it would always get an encryption of all the 0s message, whereas in the real world $\texttt{Gind-cpa}^0_{se}$ it would first get an encryption of 0, and then an encryption 1.

Claim: $\mathbf{Adv}(\mathcal{A}; \texttt{Gind-cpa}^0_{se}, \texttt{Gind-cpa}^0_{se}) =$

$$\left| \Pr\left[\mathcal{A} \to \texttt{Gind-cpa}^0_{se} = 1\right] - \Pr\left[\mathcal{A} \to \texttt{Gind-cpa}^1_{se} = 1\right] \right|$$

is non-negligible.

Success Probability

$$\begin{aligned}
&\mathbf{Adv}(\mathcal{A}; \texttt{Gind-cpa}^0_{se}, \texttt{Gind-cpa}^1_{se}) \\
&\quad = \left| \Pr\left[\mathcal{A} \to \texttt{Gind-cpa}^0_{se} = 1\right] - \Pr\left[\mathcal{A} \to \texttt{Gind-cpa}^1_{se} = 1\right] \right| \\
&\quad = \left| \Pr\left[\mathcal{A} \to \texttt{Gind-cpa}^0_{se} = 1\right] - \Pr[\mathsf{ENC}(0) = \mathsf{ENC}(0)] \right| \\
&\quad = \left| \Pr\left[\mathcal{A} \to \texttt{Gind-cpa}^0_{se} = 1\right] - 1 \right| \\
&\quad = \left| \Pr[\mathsf{ENC}(0) = \mathsf{ENC}(1)] - 1 \right| \\
&\quad = \left| 0 - 1 \right| \\
&\quad = 1
\end{aligned}$$

In the ideal game, c and c' are both encryptions of 0, and since ENC is deterministic, it produces the same ciphertext.
If the adversary interacts with the real game, the ciphertext must be different because

[1] In the literature, you might also encounter nonce-based encryption. Note that in the syntax of nonce-based encryption, the nonce takes the place of the randomness, and the syntax is then described as a deterministic function which maps the key, the message and the nonce to a ciphertext. As long as nonces don't repeat, nonce-based encryption is also a reasonable alternative way of formalizing syntax of an encryption scheme. Note that in this case, the IND-CPA definition needs to be adapted. We omit the discussion here. See `https://web.cs.ucdavis.edu/~rogaway/papers/nonce.pdf` if you are curious.

the probability that $\mathsf{ENC}(0) = \mathsf{ENC}(1)$ is 0, which follows from correctness of the encryption scheme. In a probabilistic encryption scheme, this attack would not work - since ENC would call an *se.enc* that would produce a random looking ciphertext each time (independent of previous encryptions).

Exercise 4. (Authenticated Encryption Schemes) Let $se = (se.enc, se.dec)$ be a symmetric encryption scheme, and let $m = (m.mac, m.ver)$ be a message authentication code. We construct three encryption schemes $se_a^{se,m}$ with $a \in \{\text{m+e}, \text{mte}, \text{etm}\}$ where m+e (mac-and-encrypt), mte (mac-then-encrypt), and etm (encrypt-then-mac):

$se_{\text{m+e}}.enc(k, x)$
parse $(k_m, k_{se}) \leftarrow k$
$c' \leftarrow\$\ se.enc(k_{se}, x)$
$\tau \leftarrow m.mac(k_m, x)$
$c \leftarrow (c', \tau)$
return c

$se_{\text{mte}}.enc(k, x)$
parse $(k_m, k_{se}) \leftarrow k$
$\tau \leftarrow m.mac(k_m, x)$
$x' \leftarrow (x, \tau)$
$c \leftarrow\$\ se.enc(k_{se}, x')$
return c

$se_{\text{etm}}.enc(k, x)$
parse $(k_m, k_{se}) \leftarrow k$
$c' \leftarrow\$\ se.enc(k_{se}, x)$
$\tau \leftarrow m.mac(k_m, c')$
$c \leftarrow (c', \tau)$
return c

$se_{\text{m+e}}.dec(k, c)$
parse $(k_m, k_{se}) \leftarrow k$
parse $(c', \tau) \leftarrow c$
$x \leftarrow se.dec(k_{se}, c')$
if $1 \leftarrow m.ver(k_m, x, \tau)$
  return x
else return $\perp$

$se_{\text{mte}}.dec(k, c)$
parse $(k_m, k_{se}) \leftarrow k$
$x' \leftarrow se.dec(k_{se}, c)$
parse $(x, \tau) \leftarrow x'$
if $1 \leftarrow m.ver(k_m, x, \tau)$:
  return x
else return $\perp$

$se_{\text{etm}}.dec(k, c)$
parse $(k_m, k_{se}) \leftarrow k$
parse $(c', \tau) \leftarrow c$
$x \leftarrow se.dec(k_{se}, c')$
if $1 \leftarrow m.ver(k_m, c', \tau)$:
  return x
else return $\perp$

Say for each for the following 6 statements whether you think that it is true or false. Justify your opinion: For all UNF-CMA MAC schemes m and for all IND-CPA secure symmetric encryption schemes se, it holds that...

(1) ...the encryption scheme $se_{\text{m+e}}^{se,m}$ is AE-secure.
(2) ...the encryption scheme $se_{\text{m+e}}^{se,m}$ is IND-CPA-secure.
(3) ...the encryption scheme $se_{\text{mte}}^{se,m}$ is AE-secure.
(4) ...the encryption scheme $se_{\text{mte}}^{se,m}$ is IND-CPA-secure.
(5) ...the encryption scheme $se_{\text{etm}}^{se,m}$ is AE-secure.
(6) ...the encryption scheme $se_{\text{etm}}^{se,m}$ is IND-CPA-secure.

Solution 5. Solutions:

(1) No, because we could construct an adversary that forges tags for the same plaintext. Consider a MAC scheme that computes a tag and appends the encrypted message to it $t' = t||c$. An adversary could modify c to some c' that decrypts to the same message, effectively forging t'.
(2) No. The intuition for why $se_{\text{m+e}}^{se,m}$ is not IND-CPA-secure, is that MAC only provides integrity, and not necessarily confidentiality. To show this, let us consider a MAC scheme that not only computes a tag, but appends together the message. This would satisfy UNF-CMA (integrity) but would leak the message completely.
(3) No, because an IND-CPA-secure symmetric encryption does not authenticate the ciphertext. The symmetric encryption scheme se_1, for example, allows to flip the last

bit of the ciphertext such that (a) the ciphertext is still accepted by decryption and (b) the message inside does not change. Another good intuition is to say that the message authentication code only provides integrity for the *plaintext*, but not integrity for the *ciphertext*.

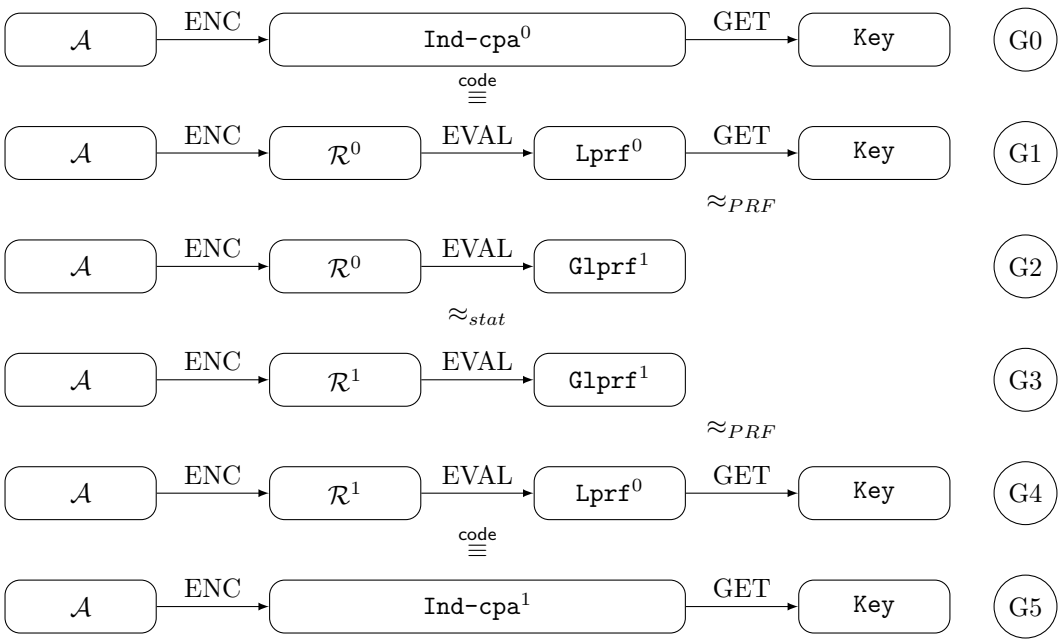
(4) Yes, provided that the tag of the message authentication code is fixed-length, because then, we encrypt some function of the message whose length does not reveal anything about the content of the original message. If the tag of the message authentication code is variable-length, then the length of the ciphertext might leak some information.
(5) Yes, because *se* provides confidentiality and m only operates on the ciphertext.
(6) Yes, because *se* provides confidentiality and m authenticates the *ciphertext*.

Exercise 5. (Constructing secure encryption from PRF) (Advanced) Let f be a secure $(\lambda, *)$-PRF. Consider the candidate encryption scheme *se*.

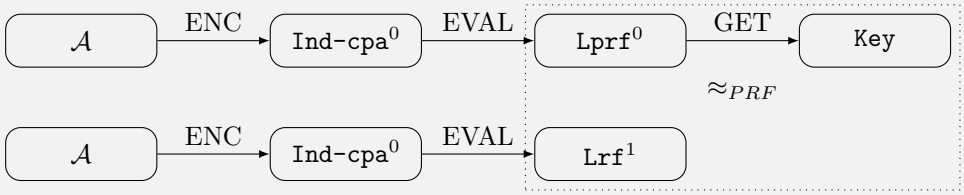
$se.enc(k, m)$	$se.dec(k, c)$
$r \leftarrow\!\!\$\ \{0,1\}^\lambda$	**parse** $(c', r) \leftarrow c$
$r' \leftarrow f(k, r, \lvert m \rvert)$	$r' \leftarrow f(k, r, \lvert c' \rvert)$
$c \leftarrow ((r' \oplus m), r)$	$m \leftarrow c' \oplus r'$
return c	**return** m

Show (via reduction) that *se* is an IND-CPA-secure encryption scheme. Provide pseudocode for your reduction and show that the probability of winning the `IND-CPA` game is negligible. See Security Definition 7 in the DAF (p. 37) for the definition of $\texttt{Lprf}^0$ and $\texttt{Glprf}^1_f$.

Hint: The following graphs show the intermediate steps that proof that the real game is indistinguishable from the ideal game. Feel free to skip some of the steps if you are stuck.



Solution 6. First, the scheme is clearly efficient (if the PRF is efficient). For correctness we consider $se.dec(k, se.enc(k, m))$ for all k, m. This reduces to $c \oplus r' = (m \oplus r') \oplus r' = m$ and correctness therefore holds.
It therefore remains to show security. We follow the outline given in the exercise and show that the advantage for each of the steps is neglegible.

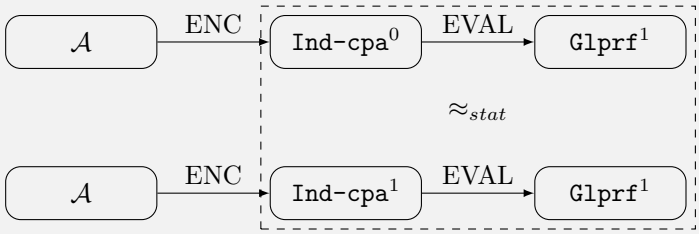


Lemma 7 (G1 and G2 are indistinguishable). *Let f be a secure PRF. Then $|\Pr[1 \leftarrow \mathcal{A} \rightarrow \texttt{Gind-cpa}^0 \rightarrow \texttt{Lprf}^0 \rightarrow \texttt{Key}] - \Pr[1 \leftarrow \mathcal{A} \rightarrow \texttt{Gind-cpa}^0 \rightarrow \texttt{Glprf}^1]|$ is negl.*

Proof. Assume it is not. Then $\mathcal{A}^0_{PRF} := \mathcal{A} \rightarrow \texttt{Gind-cpa}^0$ is a PRF-adversary with non-negl advantage:

$$\begin{aligned}\mathbf{Adv}(\mathcal{A}^0_{PRF}; \texttt{Lprf}^0 \rightarrow \texttt{Key}, \texttt{Glrf}^1) &= \mathbf{Adv}(\mathcal{A} \rightarrow \texttt{Gind-cpa}^0; \texttt{Lprf}^0 \rightarrow \texttt{Key}, \texttt{Glprf}^1) \\ &= \big|\Pr[1 \leftarrow \mathcal{A} \rightarrow \texttt{Gind-cpa}^0 \rightarrow \texttt{Lprf}^0 \rightarrow \texttt{Key}] \\ &\quad - \Pr[1 \leftarrow \mathcal{A} \rightarrow \texttt{Gind-cpa}^0 \rightarrow \texttt{Glprf}^1]\big|\end{aligned}$$

□



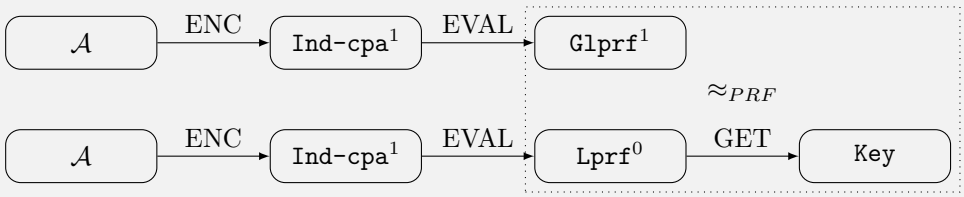
Lemma 8 (G2 and G3 are indistinguishable). *Let f be a secure PRF. Then $|\Pr[1 \leftarrow \mathcal{A} \rightarrow \texttt{Ind-cpa}^0 \rightarrow \texttt{Glprf}^1] - \Pr[1 \leftarrow \mathcal{A} \rightarrow \texttt{Ind-cpa}^1 \rightarrow \texttt{Glprf}^1]|$ is negl.*

Now it seems like this step is trivial. As we have an ideal prf, r' is a random value and so is $r' \oplus m$ – independent of m. However there's one catch: If the adversary sees more than one message with the same r (it sees r as part of the ciphertext), then encryption is "deterministic" for these messages and the adversary from exercise 3 works.
We therefore need to bound the probability that the adversary will get more than one message with the same r.

Proof. We observe, that $\mathcal{A}$ can distinguish G2 from G3 if it receives ciphertexts with the same r. As r is sampled uniformly random we can apply the birthday bound for an adversary issuing q queries to the ENC oracle.

$$\mathbf{Adv}(\mathcal{A}, \texttt{Ind-cpa}^0 \rightarrow \texttt{Glprf}^1, \texttt{Ind-cpa}^1 \rightarrow \texttt{Glprf}^1) \leq \frac{q^2}{2^\lambda}$$

As $\mathcal{A}$ is polynomially bound, it can do at most polynomially many queries making the advantage negl. □



Lemma 9 (G3 and G4 are indistinguishable). *Let f be a secure PRF. Then* $|\Pr[1 \leftarrow \mathcal{A} \to \texttt{Ind-cpa}^1 \to \texttt{Glprf}^1] - \Pr[1 \leftarrow \mathcal{A} \to \texttt{Ind-cpa}^1 \to \texttt{Lprf}^0 \to \texttt{Key}]|$ *is negl.*

Proof. Assume it is not. Then $\mathcal{A}^1_{PRF} := \mathcal{A} \to \texttt{Ind-cpa}^1$ is a PRF-adversary with non-negl advantage:

$$\begin{aligned}\mathbf{Adv}(\mathcal{A}^1_{PRF}; \texttt{Lprf}^0 \to \texttt{Key}, \texttt{Prf}^1) &= \mathbf{Adv}(\mathcal{A} \to \texttt{Ind-cpa}^1; \texttt{Lprf}^0 \to \texttt{Key}, \texttt{Glprf}^1) \\ &= \big|\Pr[1 \leftarrow \mathcal{A} \to \texttt{Ind-cpa}^1 \to \texttt{Lprf}^0 \to \texttt{Key}] \\ &\quad - \Pr[1 \leftarrow \mathcal{A} \to \texttt{Ind-cpa}^1 \to \texttt{Glprf}^1]\big|\end{aligned}$$

□

Theorem 10. *Let f be a secure PRF. Then* $|\Pr[1 \leftarrow \mathcal{A} \to \texttt{Gind-cpa}^0 \to \texttt{Prf}^0 \to \texttt{Key}] - \Pr[1 \leftarrow \mathcal{A} \to \texttt{Gind-cpa}^1 \to \texttt{Prf}^0 \to \texttt{Key}]|$ *is negligible.*

Proof. By the triangle inequality,

$$\begin{aligned}\mathbf{Adv}(\mathcal{A}; \texttt{Ind-cpa}^0 \to \texttt{Lprf}^0 \to \texttt{Key}, &\texttt{Ind-cpa}^1 \to \texttt{Lprf}^0 \to \texttt{Key}) \leq \\ &\mathbf{Adv}(\mathcal{A}; \texttt{Ind-cpa}^0 \to \texttt{Lprf}^0 \to \texttt{Key}, \texttt{Ind-cpa}^0 \to \texttt{Glprf}^1) \\ &+ \mathbf{Adv}(\mathcal{A}; \texttt{Ind-cpa}^0 \to \texttt{Glprf}^1, \texttt{Ind-cpa}^1 \to \texttt{Glprf}^1) \\ &+ \mathbf{Adv}(\mathcal{A}; \texttt{Ind-cpa}^1 \to \texttt{Glprf}^1, \texttt{Ind-cpa}^1 \to \texttt{Lprf}^0 \to \texttt{Key})\end{aligned}$$

Which is a sum of negligible functions (via lemmas 7 to 9) and therefore negligible. □