Computer Science and Engineering, UCSD | Fall 09
CSE 207: Modern Cryptography | **Instructor:** Mihir Bellare
Problem Set 5 Solutions | November 2, 2009

Problem Set 5 Solutions

Problem 1. [40 points] Let E denote AES. Let $\mathcal{K}$ be the key generation algorithm that returns a random 128-bit AES key K, and let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the symmetric encryption scheme whose encryption and decryption algorithms are as follows:

```
algorithm $\mathcal{E}_K(M)$
    if $|M| \neq 512$ then return $\perp$
    $M[1] \ldots M[4] \leftarrow M$
    $C_e[0] \xleftarrow{\$} \{0,1\}^{128}$ ; $C_m[0] \leftarrow 0^{128}$
    for $i = 1, \ldots, 4$ do
        $C_e[i] \leftarrow E_K(C_e[i-1] \oplus M[i])$
        $C_m[i] \leftarrow E_K(C_m[i-1] \oplus M[i])$
    $C_e \leftarrow C_e[0]C_e[1]C_e[2]C_e[3]C_e[4]$
    $T \leftarrow C_m[4]$
    return $(C_e, T)$
```

```
algorithm $\mathcal{D}_K((C_e, T))$
    if $|C_e| \neq 640$ then return $\perp$
    $C_m[0] \leftarrow 0^{128}$
    for $i = 1, \ldots, 4$ do
        $M[i] \leftarrow E_K^{-1}(C_e[i]) \oplus C_e[i-1]$
        $C_m[i] \leftarrow E_K(C_m[i-1] \oplus M[i])$
    if $C_m[4] \neq T$ then return $\perp$
    return $M$
```

Above, $X[i]$ denotes the i-th 128-bit block of a string whose length is a multiple of 128, and $M[1] \ldots M[4] \leftarrow M$ means we break M into 128-bit blocks.

1. **[30 points]** For each of the following notions of security, say whether the scheme is SECURE or INSECURE and justify your answer: INT-PTXT, INT-CTXT, IND-CPA, IND-CCA.

The scheme is not INT-PTXT secure. To see this, consider the following (efficient) adversary A:

```
adversary $A$
$M \xleftarrow{\$} \{0,1\}^{512}$
$(C_e, T) \xleftarrow{\$} \mathbf{Enc}(M)$
$C'_e \leftarrow C_e$ ; $C'_e[0] \leftarrow 0^{128}$
$T' \leftarrow C_e[4]$
$d \leftarrow \mathbf{Dec}((C'_e, T'))$
```

Let message M' have first block $M'[1] = M[1] \oplus C_e[0]$ and other blocks the same as M, namely $M'[j] = M[j]$ for $j = 2, 3, 4$. We claim that $\mathcal{D}_K((C'_e, T'))$ will return M'. (Check that you see why. Use the fact that T' is the correct tag of M' so $\mathcal{D}_K$ will not reject (C'_e, T').) But $M' \neq M$ as long as $C_e[0] \neq 0^{128}$, so $\mathbf{Adv}^{\text{int-ptxt}}_{\mathcal{SE}}(A) = 1 - 2^{-128}$.

The above adversary makes one **Enc** query and one **Dec** query. There is also an attack that uses zero **Enc** queries and one **Dec** query and achieves advantage 1. Can you find it?

We know that INT-CTXT implies INT-PTXT. Since the scheme is not INT-PTXT secure, it cannot be INT-CTXT secure.

The scheme is not IND-CPA secure. To see this, consider the following (efficient) adversary A:

```
adversary A
$(C_e^1, T^1) \xleftarrow{\$} \mathbf{LR}(0^{512}, 1^{512})$
$(C_e^2, T^2) \xleftarrow{\$} \mathbf{LR}(1^{512}, 1^{512})$
if $(T^1 = T^2)$ then return 1 else return 0
```

Let $E_K^4(X)$ denote $E_K(E_K(E_K(E_K(X))))$ for all $K, X \in \{0,1\}^{128}$. In game Left$_{\mathcal{SE}}$ we have $T^1 = E_K^4(0^{128})$ and $T^2 = E_K^4(1^{128})$, but these quantities are unequal because E is a blockcipher, so A returns 0. In game Right$_{\mathcal{SE}}$ we have $T^1 = T^2$ so A returns 1. Thus $\mathbf{Adv}_{\mathcal{SE}}^{\text{ind-cpa}}(A) = 1$.

We know that IND-CCA implies IND-CPA. Since the scheme is not IND-CPA secure, it cannot be IND-CCA secure.

2. [**10 points**] Discuss this scheme from the point of view of being an Encrypt-and-MAC construction. Is it? For which choices of Encrypt and MAC? How do you reconcile your findings about its security with what we know about the security of this construction?

This looks like Encrypt-and-MAC with Encrypt being CBC\$ and MAC being the (basic) CBC-MAC. We know that CBC\$ is IND-CPA and the basic CBC-MAC is SUF-CMA on fixed (here 512) length inputs. But in fact our construct is *not* an instance of Encrypt-and-MAC. The reason is that our construct uses the same key for the Encrypt and the MAC, and Encrypt-and-MAC used different, independent keys.

Encrypt-and-MAC is always INT-PTXT secure. But our construct is not. The fact that the key was the same for the Encrypt and the MAC could be exploited in an attack. We see that using the same key can result in insecurity even when the building blocks (CBC\$, basic CBC-MAC) are, taken individually, secure.

Encrypt-and-MAC does not provide IND-CPA. This does not automatically mean that IND-CPA is absent for our scheme, for two reasons. First, as we said above, it really isn't Encrypt-and-MAC. Second, even if it was, we are using specific base schemes (CBC\$ and basic CBC-MAC). The fact that Encrypt-and-MAC does not in general provide IND-CPA does not mean it won't for specific base schemes. It does not mean it will, either. We just have to check directly. We did, and it failed.

The lesson here is to be very careful about key re-use.

Problem 2. [**40 points**] Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be an IND-CPA symmetric encryption scheme, and $\mathcal{MA} = (\mathcal{K}_m, \mathcal{T}, \mathcal{V})$ a SUF-CMA MAC. Let $\overline{\mathcal{SE}} = (\mathcal{K}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ be the symmetric encryption scheme whose algorithms are as follows:

```
Game $G_0$, $\boxed{G_1}$

procedure Initialize
$K_1 \xleftarrow{\$} \mathcal{K}_e$ ; $K_2 \xleftarrow{\$} \mathcal{K}_m$ ; $b \xleftarrow{\$} \{0,1\}$ ; $S \leftarrow \emptyset$

procedure LR($M_0, M_1$)
$C \xleftarrow{\$} \mathcal{E}(K_1, M_b)$ ; $T \xleftarrow{\$} \mathcal{T}(K_2, C)$ ; $S \leftarrow S \cup \{(C,T)\}$ ; Return $(C,T)$

procedure Dec($(C,T)$)
If $(C,T) \in S$ then return $\perp$
$M \leftarrow \perp$
If $\mathcal{V}(K_2, C, T) = 1$ then
   bad $\leftarrow$ true; $\boxed{M \leftarrow \mathcal{D}(K_1, C)}$
Return $M$

procedure Finalize($b'$)
return $(b = b')$
```

Figure 1: Game G_1 includes the boxed code and game G_0 does not.

```
algorithm $\mathcal{K}$
$K_1 \xleftarrow{\$} \mathcal{K}_e$
$K_2 \xleftarrow{\$} \mathcal{K}_m$
Return $K_1 \| K_2$
```

```
algorithm $\overline{\mathcal{E}}(K_1 \| K_2, M)$
$C \xleftarrow{\$} \mathcal{E}(K_1, M)$
$T \xleftarrow{\$} \mathcal{T}(K_2, C)$
Return $(C,T)$
```

```
algorithm $\overline{\mathcal{D}}(K_1 \| K_2, (C,T))$
If $\mathcal{V}(K_2, C, T) = 0$ then return $\perp$
$M \leftarrow \mathcal{D}(K_1, C)$
Return $M$
```

Show that $\overline{\mathcal{SE}}$ is IND-CCA by establishing the following.

Theorem: Let A be an ind-cca-adversary against $\overline{\mathcal{SE}}$ that makes at most q_e **LR** queries and at most q_d **Dec** queries. Then there is an ind-cpa-adversary $A_{\mathcal{SE}}$ and a uf-cma-adversary $A_{\mathcal{MA}}$ such that

$$\mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{SE}}}(A) \;\leq\; \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(A_{\mathcal{SE}}) + 2 \cdot \mathbf{Adv}^{\text{suf-cma}}_{\mathcal{MA}}(A_{\mathcal{MA}}) \,. \tag{1}$$

Furthermore the number of **LR** queries made by $A_{\mathcal{SE}}$ is at most q_e, the number of **Tag** queries made by $A_{\mathcal{MA}}$ is at most q_e, the number of **Verify** oracle queries made by $A_{\mathcal{MA}}$ is at most q_d, and both constructed adversaries have running time that of A plus minor overhead.

Your proof should use a game sequence that includes the games G_0, G_1 of Fig. 1.

The ind-cpa-adversary $A_{\mathcal{SE}}$ and the uf-cma-adversary $A_{\mathcal{MA}}$ are depicted in Fig. 2. Games G_0, G_1 are identical-until-bad, so

$$\begin{aligned}
\frac{1}{2} \cdot \mathbf{Adv}^{\text{ind-cca}}_{\overline{\mathcal{SE}}}(A) + \frac{1}{2} \quad &= \quad \Pr\left[G_1^A \Rightarrow \mathsf{true}\right] \\
&= \quad \Pr\left[G_0^A \Rightarrow \mathsf{true}\right] + \Pr\left[G_1^A \Rightarrow \mathsf{true}\right] - \Pr\left[G_0^A \Rightarrow \mathsf{true}\right]
\end{aligned}$$

```
Adversary $A_{\mathcal{MA}}$
    $K_1 \xleftarrow{\$} \mathcal{K}_e$ ; $b \xleftarrow{\$} \{0,1\}$ ; $S \leftarrow \emptyset$
    $b' \leftarrow A^{\mathrm{LRSIM},\mathrm{DecSIM}}$
    Return $\perp$

subroutine LRSIM($M_0, M_1$)
    $C \xleftarrow{\$} \mathcal{E}(K_1, M_b)$
    $T \leftarrow \mathbf{Tag}(C)$
    $S \leftarrow S \cup \{(C,T)\}$
    Return $(C,T)$

subroutine DecSIM($(C,T)$)
    If $(C,T) \in S$ then return $\perp$
    $M \leftarrow \perp$
    If $\mathbf{Verify}(C,T) = 1$ then $M \leftarrow \mathcal{D}(K_1, C)$
    Return $M$
```

```
Adversary $A_{\mathcal{SE}}$
    $K_2 \xleftarrow{\$} \mathcal{K}_m$
    $b' \leftarrow A^{\mathrm{LRSIM},\mathrm{DecSIM}}$
    Return $b'$

subroutine LRSIM($M_0, M_1$)
    $C \xleftarrow{\$} \mathbf{LR}(M_0, M_1)$
    $T \leftarrow \mathcal{T}(K_2, C)$
    Return $(C,T)$

subroutine DecSIM($(C,T)$)
    Return $\perp$
```

Figure 2: Adversaries for Problem 2.

$$\leq \quad \Pr\left[G_0^A \Rightarrow \mathsf{true}\right] + \Pr\left[G_0^A \text{ sets } \mathsf{bad}\right] \ .$$

To conclude we observe that

$$\Pr\left[G_0^A \Rightarrow \mathsf{true}\right] = \frac{1}{2} \cdot \mathbf{Adv}_{\mathcal{SE}}^{\text{ind-cpa}}(A_{\mathcal{SE}}) + \frac{1}{2} \qquad (2)$$

$$\Pr\left[G_0^A \text{ sets } \mathsf{bad}\right] = \mathbf{Adv}_{\mathcal{MA}}^{\text{suf-cma}}(A_{\mathcal{MA}}) \ . \qquad (3)$$