MTAT.07.003 Cryptology II
Spring 2012 / Exercise session ?? / Example Solution

Exercise (Encrypt+Authenticate=NM-CPA). *Let $\mathfrak{C} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a IND-CPA secure symmetric encryption scheme and let h be a secure message authentication code with the appropriate message and key domains. Show that the paradigm first encrypt and then authenticate*

$\mathsf{Key\text{-}Gen}$
$\mathsf{sk} \leftarrow \mathsf{Gen}$
$k \leftarrow \mathcal{K}$
return (sk, k)

$\mathsf{Auth\text{-}Enc}_{\mathsf{sk},k}(m)$
$c \leftarrow \mathsf{Enc}_{\mathsf{sk}}(m)$
$t \leftarrow h(c,k)$
return (c,t)

$\mathsf{Auth\text{-}Dec}_{\mathsf{sk},k}(c,t)$
$\mathsf{if}\ t \neq h(c,k)\ \mathsf{then}\ \textbf{\textit{return}}\ \perp$
$\mathsf{else}\ \textbf{\textit{return}}\ \mathsf{Dec}_{\mathsf{sk}}(c)$

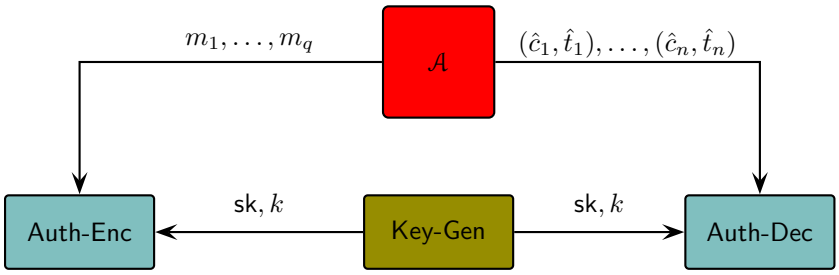
assures non-malleability under chosen message attacks (NM-CPA security).

Solution. Let the encryption scheme $\mathfrak{C}$ be (t_1, ε_1)-IND-CPA secure and let h be (t_2, ε_2)-secure message authentication code. Let us now consider the advantage of an adversary $\mathcal{A}$ against NM-CPA games:

$\mathcal{G}_0^{\mathcal{A}}$
$k \leftarrow \mathcal{K}$
$\mathsf{sk} \leftarrow \mathsf{Gen}$
$\mathcal{M}_0 \leftarrow \mathcal{A}^{\mathsf{Auth\text{-}Enc}_{\mathsf{sk},k}(\cdot)}(\mathsf{pk})$
$m \leftarrow \mathcal{M}_0, \overline{m} \leftarrow \mathcal{M}_0$
$(c,t) \leftarrow \mathsf{Enc}_{\mathsf{sk},k}(m)$
$(\pi, (\hat{c}_1, \hat{t}_1), \ldots (\hat{c}_n, \hat{t}_n)) \leftarrow \mathcal{A}^{\mathsf{Auth\text{-}Enc}_{\mathsf{sk},k}(\cdot)}(c,t)$
$\mathsf{if}\ (c,t) \in \{(\hat{c}_1, \hat{t}_1), \ldots (\hat{c}_n, \hat{t}_n)\}\ \mathsf{then}\ \textbf{return}\ 0$
$\textbf{return}\ \pi(m, \mathsf{Auth\text{-}Dec}_{\mathsf{sk},k}(\hat{c}_1), \ldots)$

$\mathcal{G}_1^{\mathcal{A}}$
$k \leftarrow \mathcal{K}$
$\mathsf{sk} \leftarrow \mathsf{Gen}$
$\mathcal{M}_0 \leftarrow \mathcal{A}^{\mathsf{Auth\text{-}Enc}_{\mathsf{sk},k}(\cdot)}(\mathsf{pk})$
$m \leftarrow \mathcal{M}_0, \overline{m} \leftarrow \mathcal{M}_0$
$(\overline{c}, \overline{t}) \leftarrow \mathsf{Auth\text{-}Enc}_{\mathsf{sk},k}(\overline{m})$
$(\pi, (\hat{c}_1, \hat{t}_1), \ldots (\hat{c}_n, \hat{t}_n)) \leftarrow \mathcal{A}^{\mathsf{Auth\text{-}Enc}_{\mathsf{sk},k}(\cdot)}(\overline{c}, \overline{t})$
$\mathsf{if}\ (\overline{c}, t) \in \{(\hat{c}_1, \hat{t}_1), \ldots (\hat{c}_n, \hat{t}_n)\}\ \mathsf{then}\ \textbf{return}\ 0$
$\textbf{return}\ \pi(m, \mathsf{Auth\text{-}Dec}_{\mathsf{sk},k}(\hat{c}_1, \hat{t}_1), \ldots)$.

Although the adversary $\mathcal{A}$ does not make an explicit call to the decryption oracle, the form of the security game implies that $\mathcal{A}$ is entitled to a singe decryption call, after which the predicate π determines the output of the game. As a result, the following interaction diagram captures interactions in the game.

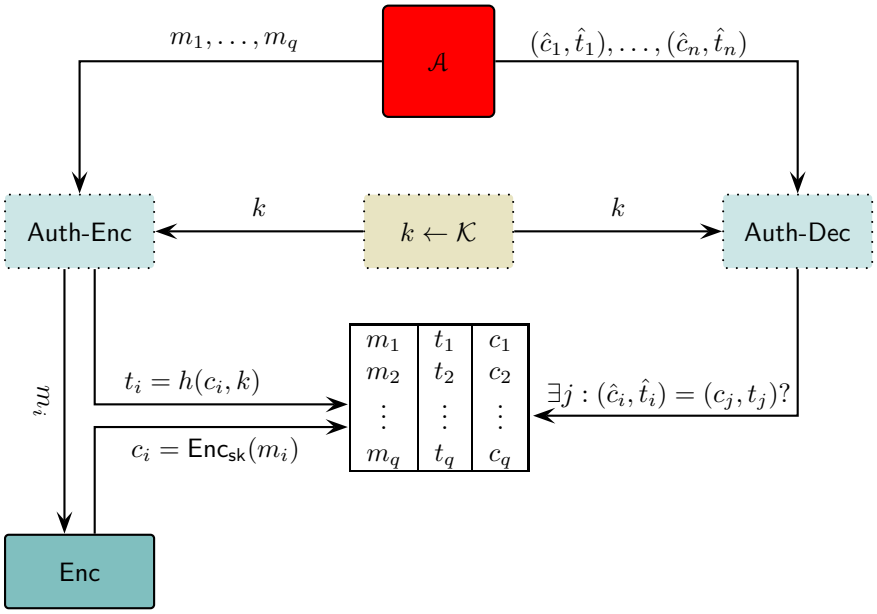


As our aim is to concert the NM-CPA adversary $\mathcal{A}$ to an IND-CPA adversary $\mathcal{B}$, we must be able to perform the decryption operation inside $\mathcal{B}$ without an access to the decryption oracle. At first glance, it seems an impossible task. Fortunately, the task is easy when the ciphertext (c_i, t_i) is legitimate ciphertext that was obtained by $\mathcal{A}$ as a reply to an encryption quary m_j. Then we know that $\mathsf{Dec}(c_i, t_i) = m_j$.

We will exploit this fact by constructing an interface that stores all encryption queries and corresponding replies as a database, which is later used for mimicking replies for the decryption calls. The schematics of

this construction is illustrated below. First, the key of the message authentication code is set up by picking $k \leftarrow \mathcal{K}$. After that $\mathcal{B}$ can simulate $\mathsf{Auth\text{-}Enc}_{\mathsf{sk},k}(m_i)$ oracle calls. To service an encryption call m_i, $\mathcal{B}$ first queries the encryption of m_i from the the encryption oracle $\mathsf{Enc}_{\mathsf{sk}}(\cdot)$ and then computes the corresponding authentication tag $t_i \leftarrow h(c_i, k)$. Next, $\mathcal{B}$ stores the triple (m_i, t_i, c_i) into the encryption table.

To decrypt pairs $(\hat{c}_1, \hat{t}_1), \ldots (\hat{c}_n, \hat{t}_n)$, the adversary $\mathcal{B}$ uses the encryption table. For each $(\hat{c}_i, \hat{t}_i)$, it tries to find matching (c_j, t_j) in the table and then use m_j as the decryption $\hat{m}_i$. If some pair $(\hat{c}_i, \hat{t}_i)$ is missing form the table, the $\mathcal{B}$ halts with zero. Otherwise $\mathcal{B}$ returns the value of $\pi(m, \hat{m}_1, \ldots, \hat{m}_n)$.



Let $\mathsf{LegitEnc}$ denote the event that $(\hat{c}_1, \hat{t}_1), \ldots (\hat{c}_n, \hat{t}_n)$ are all generated by querying the $\mathsf{Auth\text{-}Enc}_{\mathsf{sk},k}(\cdot)$ oracle. Then the event $\neg\mathsf{LegitEnc}$ does not automatically mean that the abrupt halt by $\mathcal{B}$ is different from the actual evaluation of the predicate π in the non-malleability games. In those games, the game is also halted if the decryption of $(\hat{c}_1, \hat{t}_1), \ldots, (\hat{c}_n, \hat{t}_n)$ fails. More precisely, let $\mathsf{GoodDec}$ denote the event that $\mathcal{A}$ produces a decipherable vector of ciphertexts $(\hat{c}_1, \hat{t}_1), \ldots, (\hat{c}_n, \hat{t}_n)$, i.e., $\hat{t}_i = h(\hat{c}_i, k)$ for all $i \in \{1, \ldots, n\}$. Then we can decompose the success probabilities as follows:

$$\begin{aligned}\Pr\left[\mathcal{G}_i^{\mathcal{A}} = 1\right] &= \Pr\left[\mathcal{G}_i^{\mathcal{A}} = 1 \wedge \mathsf{LegitEnc}\right] \\ &\quad + \Pr\left[\mathcal{G}_i^{\mathcal{A}} = 1 \wedge \neg\mathsf{LegitEnc} \wedge \neg\mathsf{GoodDec}\right] + \Pr\left[\mathcal{G}_i^{\mathcal{A}} = 1 \wedge \neg\mathsf{LegitEnc} \wedge \mathsf{GoodDec}\right] \quad .\end{aligned}$$

We can similarly decompose the success probabilities of $\mathcal{B}$ who plays against IND-CPA games $\mathcal{Q}_0$ and $\mathcal{Q}_1$:

$$\begin{aligned}\Pr\left[\mathcal{Q}_i^{\mathcal{B}} = 1\right] &= \Pr\left[\mathcal{Q}_i^{\mathcal{B}} = 1 \wedge \mathsf{LegitEnc}\right] \\ &\quad + \Pr\left[\mathcal{Q}_i^{\mathcal{B}} = 1 \wedge \neg\mathsf{LegitEnc} \wedge \neg\mathsf{GoodDec}\right] + \Pr\left[\mathcal{Q}_i^{\mathcal{B}} = 1 \wedge \neg\mathsf{LegitEnc} \wedge \mathsf{GoodDec}\right] \quad ,\end{aligned}$$

as the substitution of $\mathcal{B}$ leads to the game that is identical except for the last line. That is, events $\mathsf{LegitEnc}$ and $\mathsf{GoodDec}$ are well defined in games $\mathcal{Q}_0^{\mathcal{B}}$ and $\mathcal{Q}_1^{\mathcal{B}}$. By the construction

$$\Pr\left[\mathcal{G}_i^{\mathcal{A}} = 1 \wedge \mathsf{LegitEnc}\right] = \Pr\left[\mathcal{Q}_i^{\mathcal{B}} = 1 \wedge \mathsf{LegitEnc}\right]$$

as $\mathcal{B}$ obtains correct decryption values and thus the execution of the last line is identical. Analogously,

$$\Pr\left[\mathcal{G}_i^{\mathcal{A}} = 1 \wedge \neg\mathsf{LegitEnc} \wedge \neg\mathsf{GoodDec}\right] = 0 = \Pr\left[\mathcal{Q}_i^{\mathcal{B}} = 1 \wedge \neg\mathsf{LegitEnc} \wedge \neg\mathsf{GoodDec}\right] \quad ,$$

since in both games decryption failure causes the abrupt halting. Finally, we also know that

$$\Pr\left[\mathcal{Q}_0^{\mathcal{B}} = 1 \wedge \neg\mathsf{LegitEnc} \wedge \mathsf{GoodDec}\right] = \Pr\left[\mathcal{Q}_1^{\mathcal{B}} = 1 \wedge \neg\mathsf{LegitEnc} \wedge \mathsf{GoodDec}\right] \quad ,$$

since $\mathcal{B}$ halts abruptly for non-legitimately generated ciphertexts. Consequently,

$$\begin{aligned}\mathsf{Adv}_{\mathfrak{C}}^{\mathsf{ind\text{-}cpa}}(\mathcal{B}) &= \left|\Pr\left[\mathcal{Q}_0^{\mathcal{B}} = 1\right] - \Pr\left[\mathcal{Q}_0^{\mathcal{B}} = 1\right]\right| \\ &= \left|\Pr\left[\mathcal{Q}_0^{\mathcal{B}} = 1 \wedge \mathsf{LegitEnc}\right] - \Pr\left[\mathcal{Q}_1^{\mathcal{B}} = 1 \wedge \mathsf{LegitEnc}\right]\right| \quad .\end{aligned}$$

As for any event E that is defined and has identical probability in both games

$$\left|\Pr\left[\mathcal{G}_0^{\mathcal{A}} = 1 \wedge \mathsf{E}\right] - \Pr\left[\mathcal{G}_0^{\mathcal{A}} = 1 \wedge \mathsf{E}\right]\right| \leq \Pr\left[\mathsf{E}\right]$$

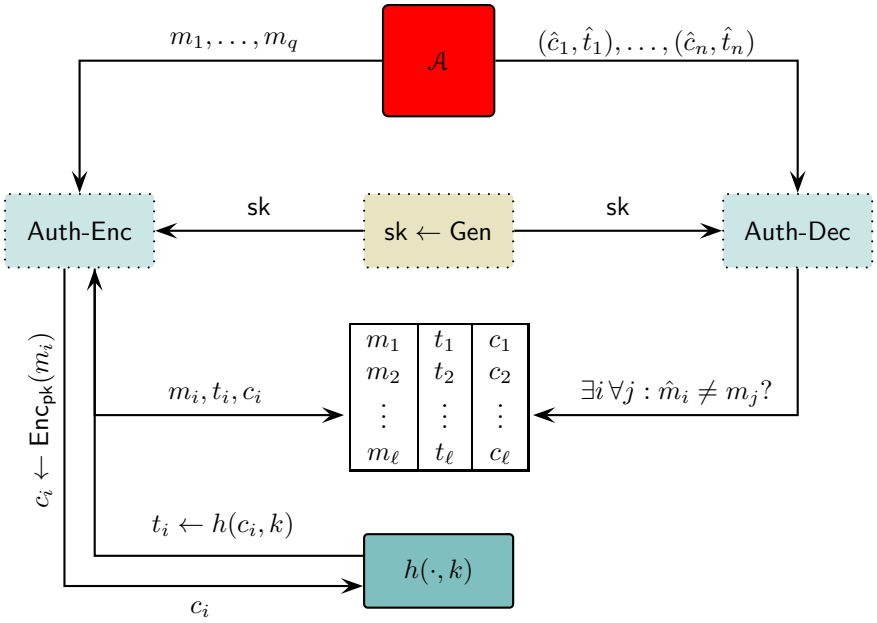
we can conclude that

$$\begin{aligned}\mathsf{Adv}_{\mathfrak{C}}^{\mathsf{nm\text{-}cpa}}(\mathcal{A}) &= \left|\Pr\left[\mathcal{G}_0^{\mathcal{A}} = 1\right] - \Pr\left[\mathcal{G}_0^{\mathcal{A}} = 1\right]\right| \\ &\leq \left|\Pr\left[\mathcal{G}_0^{\mathcal{A}} = 1 \wedge \mathsf{LegitEnc}\right] - \Pr\left[\mathcal{G}_1^{\mathcal{A}} = 1 \wedge \mathsf{LegitEnc}\right]\right| + \Pr\left[\neg\mathsf{LegitEnc} \wedge \mathsf{GoodDec}\right].\end{aligned}$$

The latter is equivalent to the inequality

$$\mathsf{Adv}_{\mathfrak{C}}^{\mathsf{nm\text{-}cpa}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathfrak{C}}^{\mathsf{ind\text{-}cpa}}(\mathcal{B}) + \Pr\left[\neg\mathsf{LegitEnc} \wedge \mathsf{GoodDec}\right] \quad .$$

To complete the proof, we need to asses the second term. Note that if the adversary $\mathcal{A}$ can create a decipherable vector $(\hat{c}_1, \hat{t}_1), \ldots, (\hat{c}_n, \hat{t}_n)$ without computing all ciphertexts legitimately, then there must exist $\hat{c}_i$ for which the authentication tag $\hat{t}_i$ was not not computed by the encryption oracle. This leads to new adversary construction $\mathcal{C}$ that simulates $\mathcal{G}_0$ to the $\mathcal{A}$ in order to get valid MAC forgery.



In this construction, the secret key sk is generated by $\mathcal{C}$ who simulates encryption and decryption queries for the $\mathcal{A}$. For that, $\mathcal{C}$ must occasionally query hash values from the hash oracle $h(\cdot, k)$ present in the security game defining security of message authentication codes. The adversary $\mathcal{C}$ runs the execution until $\mathcal{A}$ finishes. After that $\mathcal{C}$ observes all pairs $(\hat{c}_1, \hat{t}_1), \ldots, (\hat{c}_n, \hat{t}_n)$ and halts if the event $\mathsf{LegitEnc}$ occurs. Otherwise, there

must exist i such that $\hat{c}_i$ is not queried form the hashing oracle $h(\cdot, k)$. The pair $\hat{c}_i, \hat{t}_i$ will be the output of $\mathcal{C}$. It is easy to see that if the event $\mathsf{GoodDec}$ occurs then $\hat{t}_i = h(\hat{c}_i, k)$ and thus $\mathcal{C}$ wins the MAC game. Consequently, we have proven that

$$\Pr\left[\neg\mathsf{LegitEnc} \wedge \mathsf{GoodDec}\right] \leq \mathsf{Adv}_h^{\mathsf{mac}}(\mathcal{C}) \ .$$

Careful analysis shows that the overhead in the constructions is $\Theta(q \log q + n \log q)$. Thus for sufficiently small running time of $\mathcal{A}$ both constructions run in t-time and thus (t, ε_1)-IND-CPA security and (t, ε_2)-security of the message authentication code h assures that

$$\mathsf{Adv}_{\mathfrak{C}}^{\mathsf{nm\text{-}cpa}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathfrak{C}}^{\mathsf{ind\text{-}cpa}}(\mathcal{B}) + \mathsf{Adv}_h^{\mathsf{mac}}(\mathcal{C}) \leq \varepsilon_1 + \varepsilon_2 \ .$$