University of Maryland
CMSC456 — Introduction to Cryptography
Professor Jonathan Katz

Lecture 33

1 Hybrid Encryption

Recall our goal from last time: to get the advantages of public-key encryption (i.e., no need to share secrets in advance, etc.) and the efficiency of private-key encryption. We now show a way that two encryption schemes can be combined modularly in a provably-secure way.

Let $(K, \mathcal{E}, \mathcal{D})$ be any secure[1] public-key encryption scheme and let $(\mathcal{E}', \mathcal{D}')$ be a private-key encryption scheme secure against ciphertext-only[2] attacks. We will assume that the private-key scheme uses keys of length k and can encrypt messages of length L. (Without loss of generality, we assume that the public-key encryption scheme can be used to encrypt messages of arbitrary length — we saw last time how this can be done.) We construct a new public-key encryption scheme $(K'', \mathcal{E}'', \mathcal{D}'')$ for messages of length L as follows:

- Key generation algorithm K'' generates public key pk and secret key sk using key generation algorithm K.
- To encrypt message M using public key pk, algorithm $\mathcal{E}''$ does the following:
  1. $sk' \leftarrow \{0,1\}^k$
  2. $C \leftarrow \mathcal{E}_{pk}(sk')$
  3. $C' \leftarrow \mathcal{E}'_{sk'}(M)$
- To decrypt ciphertext (C, C'), algorithm $\mathcal{D}''$ does the following:
  1. Using secret key sk, compute $sk' = \mathcal{D}_{sk}(C)$
  2. Using sk', compute $M = \mathcal{D}'_{sk'}(C')$

It is easy to verify that decryption is always performed correctly (assuming decryption is always performed correctly in the original schemes). We now show that this construction is secure.

Theorem 1 *If $(K, \mathcal{E}, \mathcal{D})$ is a public-key encryption scheme which is (t, ϵ)-secure against ciphertext-only attacks and $(\mathcal{E}', \mathcal{D}')$ is a private-key encryption scheme which is (t, ϵ')-secure against ciphertext-only attacks, then the hybrid scheme $(K'', \mathcal{E}'', \mathcal{D}'')$ constructed above is a public-key encryption scheme which is $(t, 2\epsilon + \epsilon')$-secure against ciphertext-only attacks.*

[1] We saw last time that for public-key encryption schemes, security against ciphertext-only attacks is equivalent to security in the sense of indistinguishability, so calling a public-key encryption scheme "secure" is enough. For the concrete security analysis in Theorem 1, we will be more careful in our terminology.

[2] For private-key encryption schemes, we need to more carefully specify the necessary type of security.

Before giving the proof, we note the following immediate corollary (make sure you understand why this corollary follows):

Corollary 1 *Under the same assumptions as in the previous theorem, the hybrid public-key encryption scheme $(K'', \mathcal{E}'', \mathcal{D}'')$ is $(t, \ell(2\epsilon + \epsilon'))$-secure in the sense of left-or-right indistinguishability, where an adversary may query the* LR *oracle at most ℓ times.*

The corollary is a little surprising because we obtain a public-key encryption scheme secure in the sense of indistinguishability even though we started with a private-key scheme secure only against ciphertext-only attacks (which, for private-key encryption, is not equivalent to security in the sense of indistinguishability)! The reason for this is that the key for the private-key scheme is chosen newly at random every time a new message is encrypted. Thus, an adversary cannot "force" the scheme to encrypt more than one message with any particular secret key sk' (even though the adversary has access to the LR oracle for the public-key scheme).

Proof (of Theorem 1) For any algorithm A and messages m_0, m_1, we are interested in bounding:

$$\left|\Pr[A(pk, \mathcal{E}''_{pk}(m_0)) = 1] - \Pr[A(pk, \mathcal{E}''_{pk}(m_1)) = 1]\right|.$$

(The probabilities above are taken over random choice of public key pk output by K'' [in addition to the randomness for A itself], but for brevity we omit this from the description of the experiment.) Using the definition of $\mathcal{E}''$, this expression is equivalent to:

$$\Big|\Pr[sk' \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_0)) = 1]$$
$$- \Pr[sk' \leftarrow \{0,1\}^k : A(pk, A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_1)) = 1]\Big|. \qquad (1)$$

As an attempt at a proof, it seems natural to try to use algorithm A, above, to construct an algorithm A' breaking the private-key encryption scheme $(\mathcal{E}', \mathcal{D}')$. Let's see how this might work. We construct A' (who is given a ciphertext C' which is either an encryption of m_0 or m_1) as follows:

$A'(C')$
$(pk, sk) \leftarrow K$
$\widehat{sk'} \leftarrow \{0,1\}^k$
$C \leftarrow \mathcal{E}_{pk}(\widehat{sk'})$
Run $A(pk, C, C')$ and output whatever A outputs

At first glance, this seems like a perfect simulation of the view expected by A. Let's analyze the success probability of A' more carefully, however:

$$\begin{aligned}
&\left|\Pr[A'(\mathcal{E}'_{sk'}(m_0)) = 1] - \Pr[A'(\mathcal{E}'_{sk'}(m_1)) = 1]\right| \\
= &\ \Big|\Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_0)) = 1] \\
&\ - \Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_1)) = 1]\Big| \\
\neq &\ \Big|\Pr[sk' \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_0)) = 1] \\
&\ - \Pr[sk' \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_1)) = 1]\Big|,
\end{aligned}$$

where these expressions are different because, except with probability 2^{-k}, we have $sk' \neq \widehat{sk'}$. Why are these different in our experiment involving A'? Note that A' *does not know* the key under which C' was encrypted (indeed, it cannot know this key since it is attacking a private-key encryption scheme). Thus, there is no way to "fix" our definition of A' above — there is simply no way that A' can choose key sk' so that it will be equal to the key used to encrypt C'. The expressions are not equal, and we therefore need to work harder to get a proof of security.

Instead, we will construct a sequence of algorithms and bound each of their success probabilities. We first construct an A' attacking the *public-key* encryption scheme $(K, \mathcal{E}, \mathcal{D})$. We define A_1 (who is given a ciphertext C which is either an encryption of sk' or $\widehat{sk'}$ [for random $sk', \widehat{sk'}$]) as follows:

$$\begin{array}{l} A_1(pk, C) \\ \quad C' \leftarrow \mathcal{E}'_{sk'}(m_0) \\ \quad \text{Run } A(pk, C, C') \text{ and output whatever } A \text{ outputs} \end{array}$$

Let's analyze the success probability of A_1:

$$\begin{aligned} & \left| \Pr[A_1(pk, \mathcal{E}_{pk}(sk')) = 1] - \Pr[A_1(pk, \mathcal{E}_{pk}(\widehat{sk'})) = 1] \right| \\ = & \left| \Pr[sk' \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_0)) = 1] \right. \\ & \left. - \Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_0)) = 1] \right| \leq \epsilon, \end{aligned} \tag{2}$$

where the final inequality holds because we assume that $(K, \mathcal{E}, \mathcal{D})$ is (t, ϵ)-secure.

Next, we construct algorithm A_2 attacking the private-key scheme $(\mathcal{E}', \mathcal{D}')$. We define A_2 (which is given a ciphertext C' that is either an encryption of m_0 or m_1) as follows:

$$\begin{array}{l} A_2(C') \\ \quad (pk, sk) \leftarrow K \\ \quad \widehat{sk'} \leftarrow \{0,1\}^k \\ \quad C \leftarrow \mathcal{E}_{pk}(\widehat{sk'}) \\ \quad \text{Run } A(pk, C, C') \text{ and output whatever } A \text{ outputs} \end{array}$$

This algorithm is exactly the same as algorithm A', above. As before, then, the advantage of A_2 is

$$\begin{aligned} & \left| \Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_0)) = 1] \right. \\ & \left. - \Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_1)) = 1] \right| \leq \epsilon', \end{aligned} \tag{3}$$

where the final inequality is now due to the security of $(\mathcal{E}', \mathcal{D}')$ as a private-key encryption scheme.

Finally, we construct algorithm A_3 attacking the public-key scheme $(K, \mathcal{E}, \mathcal{D})$ and defined as follows (A_3 is now given a ciphertext C that is either an encryption of sk' or $\widehat{sk'}$ [for random $sk', \widehat{sk'}$]):

$A_3(pk, C)$
$C' \leftarrow \mathcal{E}'_{sk'}(m_1)$
Run $A(pk, C, C')$ and output whatever A outputs

Analyzing this in the same was as we did for A_1 (which is the same as A_3 except for one key difference), we obtain

$$\begin{aligned}
&\left|\Pr[A_3(pk, \mathcal{E}_{pk}(sk')) = 1] - \Pr[A_3(pk, \mathcal{E}_{pk}(\widehat{sk'})) = 1]\right| \\
= \;& \Big|\Pr[sk' \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_1)) = 1] \\
& - \Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_1)) = 1]\Big| \leq \epsilon. \qquad (4)
\end{aligned}$$

At this point, we step back and try to recall our original goal! We wanted to upper-bound expression (1), and we can now do so as follows (using equations (2)–(4)):

$$\begin{aligned}
& \Big|\Pr[sk' \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_0)) = 1] \\
& - \Pr[sk' \leftarrow \{0,1\}^k : A(pk, A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_1)) = 1]\Big| \\
= \;& \Big|\Pr[sk' \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_0)) = 1] \\
& - \Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_0)) = 1] \\
& + \Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_0)) = 1] \\
& - \Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_1)) = 1] \\
& + \Pr[sk', \widehat{sk'} \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(\widehat{sk'}), \mathcal{E}'_{sk'}(m_1)) = 1] \\
& - \Pr[sk' \leftarrow \{0,1\}^k : A(pk, \mathcal{E}_{pk}(sk'), \mathcal{E}'_{sk'}(m_1)) = 1]\Big| \\
\leq \;& \epsilon + \epsilon' + \epsilon = 2\epsilon + \epsilon',
\end{aligned}$$

completing the theorem. ■