

Problem Set 7 — Solutions

1. This is a simple extension of the attack described on page 340. An adversary given pk and a ciphertext c can compute $c_i := \text{Enc}_{pk}(m_i)$ for $i = 1$ to \mathcal{L} and then compare c to these \mathcal{L} possibilities. If $c = c_i$, then the adversary learns that the message was m_i .
2. Given a 2-round key-exchange protocol Π' , we construct a public-key encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ as follows:
 - $\text{Gen}(1^n)$ runs the first round of protocol Π' to obtain a first-round message A and some internal state s . The public key is A and the secret key is s .
 - $\text{Enc}_{pk}(m)$ runs the second round of Π' , given first-round message A (the second-round message may depend on the first-round message in general, even though this is not the case for Diffie-Hellman key exchange). This results in a second-round message B and a key $k \in \{0, 1\}^n$. Output the ciphertext $\langle B, k \oplus m \rangle$.
 - $\text{Dec}_{sk}(\langle B, c \rangle)$ completes running protocol Π' using the internal state s and the “second-round message” B . This results in a key $k \in \{0, 1\}^n$. The output is the message $m := k \oplus c$.

We now analyze security of Π . Fix some polynomial-time adversary \mathcal{A} . Consider as a “mental experiment” the encryption scheme $\tilde{\Pi}$ where encryption of a message m using public key pk is done by computing B as above but then choosing a *random* $\tilde{k} \in \{0, 1\}^n$ and outputting the ciphertext $\langle B, \tilde{k} \oplus m \rangle$. (Decryption is not possible, but this is ok as far as experiment PubK is concerned.) Since B is independent of the message m , and the second component of the ciphertext is just a one-time pad encryption of m , we clearly have $\Pr[\text{PubK}_{\mathcal{A}, \tilde{\Pi}}^{\text{eav}}(n) = 1] = \frac{1}{2}$.

Now construct the following adversary \mathcal{A}' attacking the key-exchange protocol Π' :

- (a) Given a transcript (A, B) and a key \hat{k} , set $pk = A$.
- (b) Run $\mathcal{A}(pk)$ until it outputs two messages m_0, m_1 .
- (c) Choose random $b \in \{0, 1\}$ and given \mathcal{A} the ciphertext $\langle B, \hat{k} \oplus m_b \rangle$.
- (d) When \mathcal{A} outputs a bit b' , output “1” iff $b' = b$.

We have (cf. Definition 9.1):

$$\begin{aligned}
 & \Pr[\text{KE}_{\mathcal{A}', \Pi'}^{\text{eav}}(n) = 1] \\
 &= \frac{1}{2} \cdot \Pr[\mathcal{A}' \text{ outputs } 1 \mid \hat{k} \text{ corresponds to } (A, B)] + \frac{1}{2} \cdot \Pr[\mathcal{A}' \text{ outputs } 0 \mid \hat{k} \text{ random}] \\
 &= \frac{1}{2} \cdot \Pr[\mathcal{A}' \text{ outputs } 1 \mid \hat{k} \text{ corresponds to } (A, B)]
 \end{aligned}$$

$$\begin{aligned}
& + \frac{1}{2} \cdot \left(1 - \Pr \left[\mathcal{A}' \text{ outputs } 1 \mid \hat{k} \text{ random} \right] \right) \\
= & \frac{1}{2} \cdot \Pr \left[b = b' \mid \hat{k} \text{ corresponds to } (A, B) \right] + \frac{1}{2} \cdot \left(1 - \Pr \left[b = b' \mid \hat{k} \text{ random} \right] \right). \quad (1)
\end{aligned}$$

Furthermore, since \mathcal{A}' runs in polynomial time and Π' is a secure key-exchange protocol, we have

$$\Pr \left[\text{KE}_{\mathcal{A}', \Pi'}^{\text{eav}}(n) = 1 \right] \leq \frac{1}{2} + \text{negl}(n) \quad (2)$$

for some negligible function negl .

Now, when \hat{k} is the key corresponding to (A, B) then it is exactly as if \mathcal{A} is being run in experiment $\text{PubK}_{\mathcal{A}, \Pi}^{\text{eav}}(n)$. Therefore,

$$\Pr \left[b = b' \mid \hat{k} \text{ corresponds to } (A, B) \right] = \Pr \left[\text{PubK}_{\mathcal{A}, \Pi}^{\text{eav}}(n) = 1 \right]. \quad (3)$$

On the other hand, when \hat{k} is random then it is exactly as if \mathcal{A} is being run in experiment $\text{PubK}_{\mathcal{A}, \tilde{\Pi}}^{\text{eav}}(n)$. Therefore,

$$\Pr \left[b = b' \mid \hat{k} \text{ random} \right] = \Pr \left[\text{PubK}_{\mathcal{A}, \tilde{\Pi}}^{\text{eav}}(n) = 1 \right] = \frac{1}{2}. \quad (4)$$

Combining Equations (1)–(4) we obtain

$$\frac{1}{2} + \frac{1}{2} \cdot \left(\Pr \left[\text{PubK}_{\mathcal{A}, \Pi}^{\text{eav}}(n) = 1 \right] - \frac{1}{2} \right) \leq \frac{1}{2} + \text{negl}(n),$$

which simplifies to

$$\Pr \left[\text{PubK}_{\mathcal{A}, \Pi}^{\text{eav}}(n) = 1 \right] \leq \frac{1}{2} + 2 \cdot \text{negl}(n).$$

Since \mathcal{A} was arbitrary, this proves that Π is secure.

3. The observation is that instead of encrypting a key k using the “pseudorandom value” h^r (cf. Lemma 10.20 and Theorem 10.22), we can just use h^r as a key itself. I.e., to encrypt a message m just output the ciphertext $\langle g^r, \text{Enc}_{h^r}(m) \rangle$. (Technically, h^r is a group element and not a bit-string so we would have to hash h^r first.)
4. The approach referred to in Proposition 10.11 would, e.g., encrypt the message $m = m_1 \| m_2$ with the ciphertext $c = c_1 \| c_2$ where $c_i = \text{Enc}_{pk}(m_i)$. But then there is an easy chosen-ciphertext attack: given a “challenge” ciphertext $c = c_1 \| c_2$, submit the ciphertext $c' = c_1 \| \text{Enc}_{pk}(0)$. (Note that $c' \neq c$, except possibly with negligible probability.) The decryption of this ciphertext will reveal the first half of m .

On the other hand, Theorem 10.10 does hold for CCA security as well. (Actually, we never formally defined the notion of indistinguishable multiple encryptions under chosen-ciphertext attacks. The notion is as one would expect, with the only subtlety being that the adversary cannot submit to the decryption oracle any of the ciphertexts it has received.) The reason is that in the setting of chosen-ciphertext attacks, one encryption of a long message (done block-by-block) is not equivalent to multiple encryptions of short messages because of the restrictions imposed on the adversary’s access to the decryption oracle.

5. The attack is as follows: the adversary outputs the two messages $m_0 = 0^{\lceil N/2 \rceil}$ and $m_1 = 1^{\lceil N/2 \rceil}$. It receives in return a ciphertext c . Then it submits the ciphertext $c' = [2^e \cdot c \bmod N]$ to its decryption oracle. If decryption succeeds (regardless of what is returned), the adversary outputs 0; otherwise it outputs 1.

Notice that if $c = \bar{m}^e \bmod N$ then $c' = (2\bar{m})^e \bmod N$. Since \bar{m} has 0s in the high-order bits, the value $2\bar{m}$ is just \bar{m} shifted left one position. Let $r = r_1 \cdots r_\ell$ denote the random string r used when encrypting c , and observe the following:

- If $m = m_1$, then decryption of c' will always fail. This is because the left-most bit of m_1 is 1, and so $2\bar{m}$ will have the form $(0^{k-1}r_1 \parallel r_2 \cdots r_\ell 0 \parallel 0^7 1 \parallel 1 \cdots 10)$. So the padding in the byte to the left of the “message” has the wrong form.
- If $m = m_0$ and $r_1 = 0$, then decryption of c' will succeed. If $r_1 = 1$ then it will not, but since each bit of r is random, we have $r_1 = 0$ with probability $1/2$.

Let b be the bit determining which message gets encrypted, and let b' denote the guess of the attack above. Analyzing the above, we have:

$$\begin{aligned} \Pr[b' = b] &= \frac{1}{2} \cdot \Pr[b' = 1 \mid b = 1] + \frac{1}{2} \cdot \Pr[b' = 0 \mid b = 0] \\ &= \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}, \end{aligned}$$

and so $\Pr[b' = b] - \frac{1}{2}$ is not negligible.

6. Because B goes second, his bit determines the XOR of the result. If B is honest, and chooses his bit at random then the result will be random.

We claim that if El Gamal encryption is used and B is dishonest, then B can make the result come out any way he wants. Say A 's ciphertext is $\langle c_1, c_2 \rangle$ where $c_1 = g^r$ and $c_2 = h^r \cdot g^a$, and a denotes the bit chosen by A . Note that B does not know r or a . Nevertheless, B can bias the result as follows:

Case 1: Say B wants to make the result 0. Then B computes $c'_1 = c_1 \cdot g$ and $c'_2 = c_2 \cdot h$ and outputs the ciphertext $\langle c'_1, c'_2 \rangle$. This is another encryption of the same bit a (and so the result will be 0) because:

$$c'_1 = c_1 \cdot g = g^r \cdot g = g^{r+1} \quad \text{and} \quad c'_2 = c_2 \cdot h = h^r \cdot g^a \cdot h = h^{r+1} \cdot g^a.$$

Case 2: Say B wants to make the result 1. Then B computes $c'_1 = (c_1)^{-1}$ and $c'_2 = (c_2)^{-1} \cdot g$. This is an encryption of $1 - a$ (and so the result will be 1) because:

$$c'_1 = (g^r)^{-1} = g^{-r} \quad \text{and} \quad c'_2 = (h^r \cdot g^a)^{-1} \cdot g = h^{-r} g^{-a} g = h^{-r} g^{1-a}.$$

The above are just examples of malleability attacks. The right type of scheme to use to prevent these attacks is a CCA-secure encryption scheme.