Modern Cryptography Jan 8, 2020

Homework 12

Lecturer: Daniel Slamanig, TA: Guillermo Perez, Karen Klein Due: 23.59 CET, Jan 15, 2020

To get credit for this homework it must be submitted no later than Wednesday, January 15th via TUWEL. If you have not registered for the tutorial (192.063 Tutorial on Introduction to Modern Cryptography 2019W) on TUWEL, please do so. If you are unable to register for the course on TUWEL for some reason, submit your homework via email to via email to `michael.walter@ist.ac.at`, please use "MC19 Homework 12" as subject.

1. Relation among Notions

   - **(4.5 Points)** Provide explicit reductions between all the consecutive security properties on slide 13 (Lecture 12). More precisely, show via reductions the following: IND-CCA2 $\implies$ IND-CCA1, IND-CCA1 $\implies$ IND-CPA, IND-CPA $\implies$ OW-CPA.

   Solution: All proofs will follow the form of contraposition ($A \implies B \iff \neg B \implies \neg A$)

   (a) IND-CCA2 $\implies$ IND-CCA1:
   Assuming an adversary $\mathcal{A}$ that breaks IND-CCA1 security of a scheme, we construct $\mathcal{A}'$ to break IND-CCA2 security. On input pk, forwards pk to $\mathcal{A}$. Any decryption queries made will be forwarded to the oracle. Note that there are no encryption queries, as $\mathcal{A}$ possesses the public key. When $\mathcal{A}$ sends (m_0, m_1), we pass this on and receive c^*, which is sent back to $\mathcal{A}$. $\mathcal{A}$ will make now make no more queries and only send b^*, which we forward to win with the same probability as $\mathcal{A}$.

   (b) IND-CCA1 $\implies$ IND-CPA:
   Analogous to above, we assume an adversary $\mathcal{A}$ that breaks IND-CPA security of a scheme with non-negl. probability. We construct $\mathcal{A}'$ to break IND-CCA1 security of the scheme. We simply forward between the two parties: pk is received and passed on to $\mathcal{A}$, who in turn sends us (m_0, m_1), which we pass on. We forward c^* to $\mathcal{A}$, who then gives us b^*, with which we win with the same (non-negl.) probability as $\mathcal{A}$. No decryption queries are used, as $\mathcal{A}$ does not have access to them.

   (c) IND-CPA $\implies$ OW-CPA:
   Assume there exists an adversary $\mathbb{A}$ that breaks OW-CPA security of a scheme. That means, on input (pk, c^*), it returns the corresponding message m^* with non-negl. probability. We construct the adversary $\mathbb{A}'$ who, on input pk, sends (m_0, m_1) and receives c^*. They then initialize $\mathbb{A}$ with (pk, c^*). $\mathbb{A}$ returns m. If $m = m_0$, they send $b^* = 0$, otherwise $b^* = 1$. As $\mathbb{A}$ returns the correct message with non-negl. probability, $\mathbb{A}'$ wins with non-negl. probability as well.

   □

2. Hybrid Encryption

- **(2 Points)** Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a CPA-secure public-key encryption scheme, and let $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ be a CCA-secure private-key encryption scheme. Consider the following construction:

Let $H : \{0,1\}^n \to \mathcal{K}'$ be a function. Construct a public-key encryption scheme as follows:

Gen^***:** on input 1^n , run $\mathsf{Gen}(1^n)$ to obtain (pk, sk). Output these as the public and private keys, respectively.

Enc^***:** on input a public key pk and a message $m \in \mathcal{M}'$, choose uniform $r_1, r_2 \in \mathcal{M}$ and output the ciphertext

$$(\mathsf{Enc}_{pk}(r_1), \mathsf{Enc}_{pk}(r_2), \mathsf{Enc}'_{H(r_1 \oplus r_2)}(m))$$

Dec^***:** on input a private key sk and a ciphertext (c_1, c_2, c_3), compute $r_1 := \mathsf{Dec}_{sk}(c_1)$, $r_2 := \mathsf{Dec}_{sk}(c_2)$ and set $k := H(r_1 \oplus r_2)$. Then output $\mathsf{Dec}'_k(c_3)$.

Is the above construction IND-CCA secure, if H is modeled as a random oracle? If yes, provide a proof. If not, provide an attack.

Solution: The scheme is not IND-CCA secure. The main observation is that for every ciphertext (c_1, c_2, c_3) we have

$$\mathsf{Dec}^*(c_1, c_2, c_3) = \mathsf{Dec}'_{H(\mathsf{Dec}_{sk}(c_1) \oplus \mathsf{Dec}_{sk}(c_2))}(c_3) = \mathsf{Dec}'_{H(\mathsf{Dec}_{sk}(c_2) \oplus \mathsf{Dec}_{sk}(c_1))}(c_3) = \mathsf{Dec}^*(c_2, c_1, c_3).$$

We can exploit this to mount an attack A on the scheme as follows. A on input of the public key queries the challenge oracle on some messages (m_0, m_1) with $m_0 \neq m_1$. The challenger responds with an encryption (c_1^*, c_2^*, c_3^*) of m_b where b is the challenge bit. If $c_1^* \neq c_2^*$ the adversary queries the decryption oracle on the ciphertext $c = (c_2^*, c_1^*, c_3^*)$. Note that $c_1^* \neq c_2^*$ implies $c \neq c^*$ and hence the query is answered with the decryption m of c. By the equation above we have $m = m_b$. Hence in this case A is able to determine b with probability 1. If, on the other hand, $c_1^* = c_2^*$ then A outputs an random guess b^* for b, which is correct with probability $1/2$.

To compute A's advantage we bound the probability of the event $\{c_1^* = c_2^*\}$. Since the scheme satisfies correctness it is not possible that there exist $r_1 \neq r_2$ such that $\mathsf{Enc}_{pk}(r_1) = \mathsf{Enc}_{pk}(r_2)$. Denote by r_1^* and r_2^* the two random messages sampled during the encryption of m_b. We hence can bound the probability of $\{c_1^* = c_2^*\}$ occurring by $\Pr[c_1^* = c_2^*] \leq \Pr[r_1^* = r_2^*] \leq 1/|M|$ where M is the message space of Π. We obtain the following bound on the success probability

of A;

$$\begin{aligned}
&\Pr[\mathsf{PubK}^{\mathsf{cca}}_{\mathsf{A},\Pi^*}(n) = 1] \\
=&\Pr[\mathsf{PubK}^{\mathsf{cca}}_{\mathsf{A},\Pi^*}(n) = 1 \mid c_1^* \neq c_2^*] \Pr[c_1^* \neq c_2^*] + \Pr[\mathsf{PubK}^{\mathsf{cca}}_{\mathsf{A},\Pi^*}(n) = 1 \mid c_1^* = c_2^*] \Pr[c_1^* = c_2^*] \\
=&1 \cdot \Pr[c_1^* \neq c_2^*] + 1/2 \cdot \Pr[c_1^* = c_2^*] \\
=&1 \cdot (1 - \Pr[c_1^* = c_2^*]) + 1/2 \cdot \Pr[c_1^* = c_2^*] \\
=&1 - 1/2 \cdot \Pr[c_1^* = c_2^*] \\
\geq&1 - 1/2 \cdot \Pr[r_1^* = r_2^*] \\
\geq&1 - 1/2 \cdot 1/|M| \\
\geq&1 - 1/2 \cdot 1/2 = 3/4 \geq 1/2 + \mathsf{negl}(\mathsf{n}),
\end{aligned}$$

showing that Π^* is not CCA-secure. In the third to last inequality we used that $|M|$ has size at least 2.

□

3. RSA Encryption

   - **[11.14 in book, 2nd edition] (3.5 Points)** Consider the following modified version of padded RSA encryption: Assume messages to be encrypted have length exactly $\|N\|/2$. To encrypt, first compute $\hat{m} := \texttt{0x00}\|r\|\texttt{0x00}\|m$ where r is a uniform string of length $\|N\|/2 - 16$. Then compute the ciphertext $c := \hat{m}^e \mod N$. When decrypting a ciphertext c, the receiver computes $\hat{m} := c^d \mod N$ and returns an error if $\hat{m}$ does not consist of `0x00` followed by $\|N\|/2 - 16$ arbitrary bits followed by `0x00`. Show that this scheme is not CCA-secure. Why is it easier to construct a chosen-ciphertext attack on this scheme than on PKCS #1 v1.5 (discussed in the lecture)?

Solution: CCA attacks on this modified scheme are easier than on PKCS #1 v1.5 because it is easier to come up with valid ciphertexts!

For the attack itself, we construct an adversary $\mathcal{A}$. We receive $pk = (N, e)$ and send $(m_0, m_1), m_0 \neq m_1$. We receive the encryption $c^* = \hat{m}^e \mod N, \hat{m} = 0\text{x}00\|r\|0\text{x}00\|m_b$ of one of our messages. We then generate a uniformly random $s \leftarrow \mathbb{Z}_N^*$ and query the decryption oracle with $c := s^e \cdot c^* \mod N = (s \cdot \hat{m})^e \mod N$. If the oracle returns a message $m = s \cdot m_b \mod N$, we divide it by s to receive m_b. This will equal one of our original messages. We then send the corresponding b and win the game with absolute certainty. If the oracle returns an error, we give up.

This clearly runs in polynomial time, but we still need to show that we produce valid padding with nong-negl. probability. Of the $2^{\|N\|}$ possible messages, $2^{\|N\|-16}$ have valid padding (as two bytes are fixed). Therefore, the probability of generating a valid message is 2^{-16}, which is constant, and thus non-negligible! □