Problem sheet 2 for Course 01410, 2023

These practice problems have the purpose of helping you understand the material better and learning the skills that are necessary to analyze cryptographic constructions, and sometimes to prepare you for the next class. All answers should be supported by a written justification. To gauge whether a justification is sufficient, ask yourself if your peers would be convinced by it without additional explanations.

Exercise 2.1 Denote by e^{OTP} and d^{OTP} the encryption and decryption algorithms of the one-time pad for n-bit strings, i.e. for $k, m, c \in \{0,1\}^n = \mathbb{K}^{\mathrm{OTP}} = \mathbb{M}^{\mathrm{OTP}} = \mathbb{C}^{\mathrm{OTP}}$ we have $e_k(m) = k \oplus m$ and $d_k(c) = k \oplus c$. Decide whether the following modifications of the one-time pad are secure. For insecure schemes, describe an adversary that wins the IND-PASS game with probability larger than $\frac{1}{2}$. For secure schemes, argue why they are secure.

For both schemes, $\mathbb{M} = \mathbb{M}^{\mathrm{OTP}}$ and $\mathbb{K} = \mathbb{C} = \{0,1\}^{2n}$, and the keys and ciphertexts consists of two parts of n bits each, $k = k_1 \| k_2$ and $c = c_1 \| c_2$ with $k_1, k_2, c_1, c_2 \in \{0,1\}^n$. For two strings $x, y \in \{0,1\}^n$, we denote by $x \odot y$ the bit-wise AND (or bit-wise product).

1. $e^1_k(m) = e^{\mathrm{OTP}}_{k_1}(m) \| (m \odot k_2)$
   $d^1_k(c) = d^{\mathrm{OTP}}_{k_1}(c_1)$.
2. $e^2_k(m) = e^{\mathrm{OTP}}_{k_1}(m) \| e^{\mathrm{OTP}}_{k_2}(m)$
   $d^2_k(c) = d^{\mathrm{OTP}}_{k_1}(c_1)$.

Note: The schemes in this exercise are of no real-world relevance and are designed for practicing the use of security definitions.

Solution:

1. (e^1, d^1) is not secure. We define an IND-PASS adversary $\mathcal{A}$ as follows. $\mathcal{A}$ sends messages $m_0 = 0^n$ and $m_1 = 1^n$ to the challenger, where 0^n here is the n-bit string where all bits are 0, and similar for 1^n. Now $\mathcal{A}$ receives a ciphertext $c = c_1 \| c_2$ from the challenger. If $c_2 = 0^n$, $\mathcal{A}$ outputs $b' = 0$ if $c_2 = 0^n$ and $b' = 1$ else.

   Now let us analyze the probability that $\mathcal{A}$ wins the IND-PASS game. The challenger has sampled $b \leftarrow \{0,1\}$ and computed $c = e_k(m_b)$. Letting $\mathcal{A}_1$ be the second stage of $\mathcal{A}$ that gets c as input and outputs b', we compute

   $$\begin{aligned}\Pr[b' = b] =& \Pr[b' = b | b = 0] \Pr[b = 0] + \Pr[b' = b | b = 1] \Pr[b = 1] \\ =& \frac{1}{2} \left(\Pr[b' = b | b = 0] + \Pr[b' = b | b = 1] \right) \\ =& \frac{1}{2} \left(\Pr[\mathcal{A}(e_k(m_0)) = 0] + \Pr[\mathcal{A}(e_k(m_1)) = 1] \right).\end{aligned}$$

   We now analyze the cases $b = 0$ and $b = 1$ separately. For $b = 0$, we have $m_0 \odot k_2 = 0^n \odot k_2 = 0^n$, so

   $$\Pr[\mathcal{A}_1(e_k(m_0)) = 0] = 1.$$

   For $b = 1$ we have $m_1 \odot k_2 = 1^n \odot k_2 = k_2^n$, so unless $k_2 = 0^n$, $\mathcal{A}_1$ outputs 1 in this case. We therefore get

   $$\begin{aligned}\Pr[\mathcal{A}_1(e_k(m_1)) = 1] =& \Pr[k_2 \neq 0^n] \\ =& 1 - 2^{-n}.\end{aligned}$$

Combining our three equations we get

$$\begin{aligned}\Pr[b' = b] =& \frac{1}{2}\left(\Pr[\mathcal{A}(e_k(m_0)) = 0] + \Pr[\mathcal{A}(e_k(m_1)) = 1]\right)\\ =& \frac{1}{2}\left(2 - 2^{-n}\right)\\ =& 1 - 2^{-(n+1)}.\end{aligned}$$

Which is very close to one and thus in particular much larger than 1/2.

2. (e^2, d^2) is secure. We know: the one-time pad is perfectly secure in the sense that the adversary does not learn anything new about the message by looking at the ciphertext, as long as a fresh, uniformly random key is used. Therefore we can argue that the adversary does not learn anything from the one-time pad encryption c_2 given prior knowledge about c_1, as c_2 is produced using a fresh, uniformly random key k_2. Thus an adversary can just as well only consider c_1. But c_1 is a one-time pad ciphertext, so that doesn't tell the adversary anything about the plaintext, either.

   In the language of Lecture 2, we can make this argument formal if desired. Let M be a random variable representing the adversary's view of a message. (Example: in the IND-PASS game, $M = m_0$ with probability 1/2, and $M = m_1$ with probability 1/2.) Let $C = C_1 \| C_2$ be the ciphertext resulting from encrypting M with a random key. Let $m, c_1, c_2 \in \{0,1\}^n$ and set $c = c_1 \| c_2$. We define a random variable M' by $\Pr[M' = m'] = \Pr[M = m' | C_1 = c_1]$ for all $m' \in \mathbb{M}$. We compute

$$\begin{aligned}\Pr[M = m | C = c] =& \Pr[M = m | C_1 = c_1 \wedge C_2 = c_2]\\ =& \Pr[M' = m | C_2 = c_2]\\ \overset{*}{=}& \Pr[M' = m]\\ =& \Pr[M = m | C_1 = c_1]\\ \overset{*}{=}& \Pr[M = m].\end{aligned}$$

   In the equations with the $*$, we have used the perfect secrecy of the one-time pad.

Exercise 2.2 We use the notation from last week's problem sheet for ASCII characters. Let $\text{ord}(c)$ be the index of an ASCII character c, i.e. $\text{ord}(c)$ is defined such that $c = y_{\text{ord}(c)}$. Define the one-time pad encryption scheme for ASCII strings as follows. For a message string m and a key string k (of equal length ℓ), the one-time pad encryption outputs $e_k(m) = c = (c_1, ..., c_\ell)$ such that $c_i = \text{ord}(m_i) \oplus \text{ord}(k_i)$, and $\oplus$ is the XOR operation when the integer values are represented in binary.

Example: For $\ell = 2$ we have the message $m =$`Hi` and the key string $k =$ `B`y_5 (the ASCII character with index 5 is not printable). Using a subscript 2 to indicate binary numbers, we have ord(`H`)$= 72 = 1001000_2$, ord(`i`)$= 105 = 1101001_2$, ord(`B`)$= 66 = 1000010_2$ and $ord(y_5) = 6 = 0000101_2$. We therefore get $e_k(m) = (10, 108)$.

The following are the ciphertexts of encrypting with the same key the ASCII strings `grape`, `apple`, `pears`, in a different order:

$c = (17, 46, 118, 127, 77)$, $c' = (0, 59, 103, 97, 91)$, $c'' = (23, 44, 103, 99, 77)$

In addition, $\tilde{c} = (0, 41, 104, 50, 9)$ is the encryption of an unknown ASCII string, using the same key as before. Break the OTP and decrypt $\tilde{c}$!

Solution: We can use the fact that $e_k(m) \oplus e_k(m') = m \oplus m'$. grape and pears both have 'a' as third letter, but apple does not. Two strings XOR to the all zero string if and only if they are equal. We therefore know that the two ciphertexts corresponding to the plaintexts pears and grape are the two with identical third component, i.e. c' and c''. Thus c must be an encryption of apple. We therfore get the key $k = y_{i_1} \| y_{i_2} \| y_{i_3} \| y_{i_4} \| y_{i_5}$ via $i_1 = c_1 \oplus \mathrm{ord}(a)$, $i_2 = c_2 \oplus \mathrm{ord}(p)$ etc. Under the resulting key, $\tilde{c}$ decrypts to pwn!!.