Tutorial 2 - Solutions

Computer Security
School of Informatics
University of Edinburgh

In this second tutorial for the Introduction to Computer Security course we cover Cryptography. The tutorial consists of questions from past years exams.

You are free to discuss these questions and their solutions with fellow students also taking the course, and also to discuss in the course forum. Bear in mind that if other people simply tell you the answers directly, you may not learn as much as you would by solving the problems for yourself; also, it may be harder for you to assess your progress with the course material.

1 Hash functions

Let $\mathcal{M} = \{0,1\}^*$ and $\mathcal{T} = \{0,1\}^n$ for some integer n.

1. Explain what does it mean for a hash function $h : \mathcal{M} \to \mathcal{T}$ to be one-way.

   Solution

   A function h is a one-way function if for all $y \in T$ there is no efficient algorithm which given y can compute x such that $h(x) = y$.

2. Explain what does it mean for a hash function $h : \mathcal{M} \to \mathcal{T}$ to be collision resistant.

   Solution

   A function h is collision resistant if there is no efficient algorithm that can find two messages m_1 and $m_2 \in M$ such that $h(m_1) = h(m_2)$.

3. Suppose $h : \mathcal{M} \to \mathcal{T}$ is collision resistant. Is h also one-way? If so, explain why. If not, give an example of a collision resistant function that is not one-way.

   Solution

   Let g be a hash function which is collision resistant and maps arbitrary-length inputs to $n-1$-bit outputs. Consider the function h defined as:

   $$h(x) = \begin{cases} 1||x & \text{if } x \text{ has bitlength } n-1 \\ 0||g(x) & \text{otherwise} \end{cases}$$

   where $||$ denotes concatenation. Then h is an n-bit hash function which is collision resistant but not one-way. As a simpler example, the identity function on fixed-length inputs is collision resistant but not one way.

4. Suppose $h : \mathcal{M} \to \mathcal{T}$ is one-way. Is h also collision resistant? If so, explain why. If not, give an example of a one-way function that is not collision resistant. Suppose the DLOG assumption is true.

Solution

Let p a large prime and g a generator of $\mathbb{Z}_p$. Let h be the function $h(x) = g^x \mod p$. This function is one way from the DLOG assumption: inverting exponentiation over a discrete group is computationally hard. However, this function is not collision resistant. Indeed, if we choose x_1 and $x_2 = x_1 + (p-1)$, then

$$g^{x_2} = g^{x_1+(p-1)} = g^{x_1} g^{(p-1)} = g^{x_1} \pmod{p} \ .$$

5. Bob is on an under cover mission for a week and wants to prove to Alice that he is alive each day of that week. He has chosen a secret random number, s, which he told to no one (not even Alice). But he did tell her the value $H = h(h(h(h(h(h(h(s)))))))$, where h is a cryptographic hash function. During that week Bob will have access to a broadcast channel, so he knows any message he sends to Alice will be received by Alice. Unfortunately Bob knows that Eve was able to intercept message H. Explain how Bob can broadcast a single message everyday that will prove to Alice that he is still alive. Note that your solution should not allow anyone (and in particular Eve) to replay any previous message from Bob as a (false) proof that he still is alive.

Solution

Let d range from 1 to 7 and denote the day of the week. On day d, Bob broadcasts message $h^{7-d}(s)$. Because of one wayness of h, from previous seen messages $h^7(s), \ldots, h^{7-(d-1)}(s)$ no one else can compute $h^{7-d}(s)$ but Bob. But anyone (and in particular Alice) can verify that $h^{7-(d-1)}(s) = h(h^{7-d}(s))$ that is the message received on day $d-1$ is the hash of the message received on day d, proving that Bob is alive.

2 Symmetric encryption

Let $(\mathcal{E}_{32}, \mathcal{D}_{32})$ be a secure (deterministic) block cipher with 32-bits key size and 32-bits message size. We want to use this cipher to build a new (deterministic) block cipher $(\mathcal{E}_{64}, \mathcal{D}_{64})$ that will encrypt 64-bits messages under 64-bits keys. We consider the following encryption algorithm. To encrypt a message M under a key K, we split M into two parts M_1 and M_2, and we also split K into two parts K_1 and K_2. The ciphertext C is then computed as $\mathcal{E}_{32}(K_1, M_1)||\mathcal{E}_{32}(K_2, M_2)$. In other words we concatenate the encryption of M_1 under K_1 using $\mathcal{E}_{32}$, with the encryption of M_2 under K_2 using $\mathcal{E}_{32}$.

1. What is the corresponding decryption algorithm? To justify your answer prove that the consistency property is satisfied.

Solution

We just split C into two parts C_1 and compute the underlying plaintext as $\mathcal{D}_{32}(K_1, C_1)||\mathcal{D}_{32}(K_2, C_2)$. The proof of consistency is trivial given the consistency of $(\mathcal{E}_{32}, \mathcal{D}_{32})$. Indeed

$$\begin{aligned}
\mathcal{D}_{64}(K_1||K_2, \mathcal{E}_{64}(K_1||K_2, M_1||M_2)) &= \mathcal{D}_{64}(K_1||K_2, \mathcal{E}_{32}(K_1, M_1)||\mathcal{E}_{32}(K_2, M_2)) \\
&= \mathcal{D}_{64}(K_1, \mathcal{E}_{32}(K_1, M_1))||\mathcal{D}_{64}(K_2, \mathcal{E}_{32}(K_2, M_2)) \\
&= M_1||M_2
\end{aligned}$$

2. Consider the following game.

   - In the first phase, the attacker chooses a few 64-bit plaintext messages M_1, …, M_n and gets back from an encryption oracle the corresponding ciphertexts C_1, …, C_n under some key K that he does not know. The attacker gets to know that C_i is the ciphertext corresponding to M_i for all $i \in \{1, \dots, n\}$.
   - In the second phase the attacker builds two 64-bit messages M_A and M_B and gets back C which is the encryption under K either of M_A or M_B. But now, the attacker doesn't know if the plaintext underlying C is M_A or M_B and has to guess it.

   Informally, a symmetric cipher is said to be vulnerable to a chosen plaintext attack if the attacker can guess (with high probability) which of M_A or M_B is the plaintext corresponding to C. Show that the new cipher $(\mathcal{E}_{64}, \mathcal{D}_{64})$ is subject to a chosen plaintext attack even though $(\mathcal{E}_{32}, \mathcal{D}_{32})$ is not.

Solution

Let $M_1 = 0^{32}||0^{32}$ and $M_2 = 1^{32}||1^{32}$. Let $C_1 = \mathcal{E}_{32}(K_1, 0^{32})||\mathcal{E}_{32}(K_2, 0^{32})$ and $C_2 = \mathcal{E}_{32}(K_1, 1^{32})||\mathcal{E}_{32}(K_2, 1^{32})$, and let $M_A = 0^{32}||1^{32}$ and $M_B = 1^{32}||0^{32}$. Given C_1 and C_2 the attacker can trivially compute $\mathcal{E}_{64}(0^{32}||1^{32}) = \mathcal{E}_{32}(K_1, 0^{32})||\mathcal{E}_{32}(K_2, 1^{32})$ and $\mathcal{E}_{64}(1^{32}||0^{32}) = \mathcal{E}_{32}(K_1, 1^{32})||\mathcal{E}_{32}(K_2, 0^{32})$, and thus win the game with probability 1. Thus this new scheme is not secure under chosen plaintext attack.

3. A symmetric cipher is said to be vulnerable to a known plaintext attack if given a plaintext message M and its corresponding ciphertext C under some key K not known to the attacker, the attacker can recover the key K in a reasonable amount of time (that is significantly less than by a brute force-attack). Show that $(\mathcal{E}_{64}, \mathcal{D}_{64})$ is vulnerable to a known plaintext attack.

Solution

(a) A brute force attack consists in trying all the possible keys in the key space. So for the new algorithm the brute-force attack has complexity 2^{64}.

(b) First, use brute-force to recover K_1. We know that $C_1 = \mathcal{E}_{32}(K_1, M_1)$, and we know M_1, C_1, so try all possibilities for K_1 and see which one is consistent with this equation. Next, use brute force to recover K_2, by a similar method. This requires $2^{32} + 2^{32} = 2^{33}$ trial decryptions in total, which is easily feasible. That means our attack has complexity 2^{33}.

3 Cryptographic Proofs

1. Prove that in a classroom of 23 students the probability that any two students have the same birthday is over 50%, a.k.a the Birthday Paradox. Suppose birthdays are distributed uniformly over the 365 days of the year.

   Solution

   Let A be the event in which there exists at least one pair of students with the same birthday. Then A' is the complementary event, in which there exists no pair of students with the same birthday. It is $P(A) = 1 - P(A')$. We can calculate $P(A')$ as follows:
   If there was only one student, then there would be no collision with probability 1. Adding a second student would make the probability of collision equal to $\frac{364}{365}$, since this is the probability of the second student having a birthday on the same day as the first student. Adding a third student would make the probability of collision equal to $\frac{364}{365} \times \frac{363}{365}$, since the birthday of the third student is independent of the first two and now two days of the year are already taken. With the same reasoning we deduce that

   $$P\left(A'\right) = \frac{364}{365} \times \frac{363}{365} \times \cdots \times \frac{365-22}{365} \approx 0.4927 \ .$$

   Thus $P(A) = 1 - P(A') \approx 0.5073$.

2. Prove that, given a collision-resistant one-way compression function $h : \{0,1\}^{2n} \rightarrow \{0,1\}^n$, the Merkle-Dåmgard construction builds a collision resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$.

   Solution

   We will use contradiction for this proof. Suppose $\exists x, x' : H(x) = H(x')$. Also let L denote the length of x and B the number of blocks x is split in. Additionally, let $x = x_1, x_2, \ldots, x_B$ and $x_{B+1} = L$. Similar definitions hold for x'. There are two cases:

   (a) $L \neq L'$. Then the last step for the calculation of H is $h(z_B||L) = z_{B+1} = z_{B'+1} = h(z_{B'}||L')$, thus we found a collision for h.

   (b) $L = L'$. Then $B = B'$, thus $x_{B+1} = x'_{B'+1}$. Seeing that there exists an earlier i such that $x_i \neq x'_i$ but $h(x_i) = h(x'_i)$ is a direct application of induction.

3. Prove that the RSA encryption scheme is consistent: Given a public - secret keypair $(n, e), d$, it is

   $$Dec_{RSA}\left(d, Enc_{RSA}\left((n,e), m\right)\right) = m \pmod{n} \ .$$

   Hint: Use Euler's theorem.

Solution

It is

$$ed = 1 \pmod{\phi(n)} \Rightarrow \exists a \in \mathbb{N} : ed = 1 + a\phi(n) \quad . \tag{1}$$

Thus

$$\begin{gathered}
Dec_{RSA}\left(d, Enc_{RSA}\left(\left(n, e\right), m\right)\right) = \\
\left(m^e\right)^d \mod n = \\
m^{ed} \mod n \overset{(1)}{=} \\
m^{a\phi(n)+1} \mod n = \\
m^{a\phi(n)} m \mod n \overset{Euler}{=} \\
m \mod n \ .
\end{gathered}$$