


دانشکده‌ی علوم ریاضی

مقدمه‌ای بر رمزنگاری تحویل اصلی: ۱۹ اردیبهشت ۱۴۰۰

تمرین شماره ۵

مدرّس: دکتر شهرام خزائی *تحویل نهایی: ۲۶ اردیبهشت ۱۴۰۰*

- Upload your answers on courseware with the name: StudentNumber.pdf
- Upload a PDF file. Image and zip formats are not accepted.
- Similar answers will not be graded.
- NO answers will be accepted via e-mail.
- You can't upload files bigger than 2 Mb, so you'd better type.
- Deadline time is always at 23:55 and will not be extended.
- You should submit your answers before soft deadline.
- You will lose 5 percent for each day delay if you submit within a week after soft deadline.
- You can not submit any time after hard deadline.
- This problem set includes 85 points.
- For any question contact Aysan Nishaburi via `aysannishaburi@gmail.com`.

Problem 1

For many block cipher encryption modes such as CBC mode, messages need to be a multiple of the block size. Messages that are not a multiple of the block size can still be encrypted, but need to be padded to a multiple of the block size. The padding moreover needs to be reversible so that the receiver can recover the original (unpadded) message when decrypting. For each of the following padding schemes, decide if the padding is reversible: that is, for any message, after padding to a multiple of the block length, it is possible to recover the message again. If the padding is reversible, explain how to recover the message and why recovery is guaranteed to work. If not, explain how it fails.

1. (5 Points) **Null Padding:** Append 0's to the message until it is a multiple of the block length

2. (5 Points) **Bit Padding, version 1:** Let N be the number of bits necessary to add to the message for it to become a multiple of the block length. If $N > 0$, append 10^{N-1} (that is, a 1 followed by $N-1$ 0's) to the message. If N = 0 (the message is already a multiple of the block length), do nothing.

3. (5 Points) **Bit Padding, version 2:** This is the same as part 2, except that in the case $N = 0$, we append an entire block, set to 10^{B-1}, where B is the block length in bits.

4. (5 Points) **PKCS7 Padding:** Assume the message is an integer number of bytes, but not an integer number of blocks. Let N be the number of bytes necessary to pad to a multiple of the block length. If $N = 0$ (which means the message is already a multiple of the block length) let N be equal to the block length (in bytes). Now pad with N bytes, each byte set to the value N. For example, if $N = 3$, append 3 bytes to the message, each byte set to 00000011.

5. (5 Points) PKCS7 padding, except that if the message is already a multiple of the block length, do not add any padding.

Problem 2

(30 Points) Let h be a collision-resistant hash-function.

1. Consider

$$h_s^0(x) = \begin{cases} h_s(x)||1 & \text{if } x_1 = 0 \\ 0^{|h_s(x)|+1} & \text{otherwise} \end{cases} \quad (1)$$

$$h_s^1(x) = \begin{cases} h_s(x)||1 & \text{if } x_1 = 1 \\ 0^{|h_s(x)|+1} & \text{otherwise} \end{cases} \tag{2}$$

$$\hat{h}_s(x) = h_s^0(x)||h_s^1(x)$$

Prove that $\hat{h}$ is collision-resistant.

2. Now let

$$h_s^a(x) := h_s(x)_1 ... h_s(x)_{\lceil \frac{|h_s(x)|}{2} \rceil}$$

$$h_s^b(x) := h_s(x)_{\lceil \frac{|h_s(x)|}{2} \rceil + 1} ... h_s(x)_{|h_s(x)|}$$

where the ith bit of a string x is denoted by x_i. Prove or disprove: At least one of h_s^a and h_s^b is collision resistant.

3. Answer part 2 in the case that the output of h_s^a and h_s^b is equal for every input x. Prove your answer.

Problem 3

(30 Points) Let (E, D) be an encryption system that provides authenticated encryption. Here E does not take a nonce as input and therefore must be a randomized encryption algorithm. Which of the following systems provide authenticated encryption? For those that do explain why. For those that do not, present an attack that either breaks CPA security or ciphertext integrity.

1. $E_1(k, m) = [c \leftarrow E(k, m), \text{ outputs } (c, c)]$ and $D_1(k, (c_1, c_2)) = D(k, c_1)$
2. $E_2(k, m) = (E(k, m), E(k, m))$ and $D_2(k, (c_1, c_2)) = \begin{cases} D(k, c_1) & \text{if } D(k, c_1) = D(k, c_2) \\ \perp & \text{otherwise} \end{cases}$

   To clarify: $E(k, m)$ is randomized so that running it twice on the same input will result in different outputs with high probability.
3. $E_3(k, m) = (E(k, m), H(m))$ and $D_3(k, (c_1, c_2)) = \begin{cases} D(k, c_1) & \text{if } H(D(k, c_1)) = c_2 \\ \perp & \text{otherwise} \end{cases}$

   where H is a collision resistant hash function.

Problem 4 (Optional)

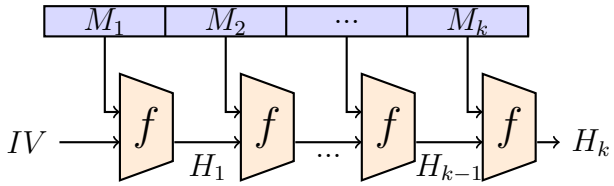
Let $h : \{0,1\}^* \to \{0,1\}^n$ be a hash function constructed by iterating a collision resistant compression function using the Merkle-Damgard construction below. The idea is to split the message M into blocks of constant length

$$M = M_1||M_2||...||M_k$$

and to process these blocks along with the intermediate hash values

$$H_1, ..., H_{k-1}$$

through the compression function f. H_k is the hash value of M, that is $h(M) = H_k$.



1. (5 Bonus Points) Show that defining $MAC_k(M) = h(k||M)$ results in an insecure MAC. That is, show that given a valid text/MAC pair (M, H) one can efficiently construct another valid text/MAC pair (M', H') without knowing the key k. Assume for simplicity that the key length is the same as the length of the message block.

2. (15 Bonus Points) Show that appending the secret key k, that is defining $MAC_k(M) = h(M||k)$ results in a MAC that isn't collision resistant.
   Recall that by definition the property of being collision resistant for MAC means that finding two messages $x \neq x'$ such that

   $$MAC_k(x) = MAC_k(x'),$$

   implies the computational effort of 2^n operations, where n is the size of MAC (and hash). Describe an attack (based on the Merkle-Damgard structure above) that uses less than 2^n operations to create the MAC forgery, a legitimate MAC value for some message x without revealing the key k.