CS60065: Cryptography and Network Security Total: 50 *points*

Assignment 3

Instructor: *Monosij Maitra* Due: *11.59 pm, Oct 6, 2024*

Note: The basic policies are stated in the course page. Using GPT (or similar tools) to solve problems from the assignment is **strictly prohibited**. Use of any other (possibly online) source(s) **must** be clearly stated in the solution. **Any dishonesty, if caught, will yield zero credits for the entire assignment.**

A. [**Block Ciphers :** $\mathbf{8 + 10 + (3 \times 4) = 30}$ **points**.]

1. Compute the linear approximation table for the following S-box:

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\sigma_S(x)$	8	4	2	1	C	6	3	D	A	5	E	7	F	B	9	0

2. This is your chance to break a cryptosystem. As we now know, building encryption is non-trivial. This problem illustrates how easy it is to weaken a strong scheme with minor modifications.

   Recall we discussed briefly in class about key-whitening that it can be a good strategy for strengthening block ciphers against brute-force attacks. DESX, in particular, provides such an approach modifying DES. In DESX, two additional 64-bit keys K_1 and K_2 are XOR-ed to the message and ciphertext, respectively, prior to and after DES scheme as follows:

$$y = \mathsf{DESX}_{\mathsf{K},\mathsf{K}_1,\mathsf{K}_2}(x) = \mathsf{DES}_{\mathsf{K}}(x \oplus \mathsf{K}_1) \oplus \mathsf{K}_2$$

   We now look at the following variant of key-whitening against DES, which we will call DESA:

$$y = \mathsf{DESA}_{\mathsf{K},\mathsf{K}'}(x) = \mathsf{DES}_{\mathsf{K}}(x) \oplus \mathsf{K}'$$

   Even though the method looks similar to key-whitening, it hardly adds to the security. Assume you have 2 valid message-ciphertext pairs under DESA. Show that breaking DESX is roughly as difficult as a brute-force attack against single DES.

3. Let $\mathsf{E}(\cdot)$ be a block cipher for encrypting 5-bit message blocks, where E is a bit permutation, which depends on the key. Assume that for a given key encryption is as follows: $\mathsf{E}(m_1m_2m_3m_4m_5) = (m_2m_5m_4m_1m_3)$. Encrypt $x = 01101\ 11011\ 11010\ 00110$ with four different modes of operation ECB, CBC, CFB and OFB, and provide the corresponding ciphertexts y. Use $\mathsf{IV} = 11001$.

B. [**Cryptographic Hash Functions & MACs:** $\mathbf{(3 + 4) + (4 + 3) + (3 + 3) = 20}$ **points**.]

1. Let $h : \mathcal{X} \to \mathcal{Y}$ be an (N, M)-hash function and let $h^{-1}(y) = \{x : h(x) = y\})$. Let $s_y = |h^{-1}(y)|$ for any $y \in \mathcal{Y}$. Recall the algorithm Preimage discussed in class used to solve the **Preimage** problem for any hash function. Consider solving **Preimage** for the function h, using Preimage, assuming that we have only oracle access for h. For a given $y \in \mathcal{Y}$, let $\mathcal{X}_0 \subseteq \mathcal{X}$ be a *random* subset of size q. Let ϵ be the success probability of Preimage, given y.

   (a) Prove that $\epsilon = 1 - \dfrac{\binom{N-s_y}{q}}{\binom{N}{q}}$.

   (b) With $q = 1$, prove that the *average* success probability of Preimage over all $y \in \mathcal{Y}$ is $\dfrac{1}{M}$.

2. Suppose g is a collision-resistant hash function that takes bitstrings of arbitrary lengths as input and produces n-bit message digests. Define a hash function h as follows:

$$h(x) = \begin{cases} 0||x & \text{if } x \text{ is a bitstring of length } n \\ 1||g(x) & \text{otherwise} \end{cases}$$

(a) Prove that h is collision-resistant.

(b) Prove that h is *not* preimage-resistant. [Show preimages for h can easily be found for half of the possible message digests.]

3. Assume that the hash family $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$ provides a secure MAC algorithm, where the tag for a message $x \in \mathcal{X}$ is computed as $h_K(x)$ for any $h \in \mathcal{H}, K \in \mathcal{K}$.

   (a) Suppose we compute the tag as $x||h_K(x)$. Is this new MAC still secure or not? Explain.

   (b) Discuss why the general strategy of $\mathsf{MAC\text{-}and\text{-}Encrypt}$ should be avoided. [Modify a secure MAC algorithm and examine the impact of this change in the context of $\mathsf{MAC\text{-}and\text{-}Encrypt}$.]