University of Maryland
CMSC456 — Introduction to Cryptography
Professor Jonathan Katz

Problem Set 2

Due at the *beginning* of class on Oct. 5

1. In the one-time pad encryption scheme for messages of length ℓ, it can sometimes happen that the key is the all-zero string (i.e., the key is 0^ℓ). In this case, the encryption of a message m is given by $m \oplus 0^\ell = m$ and therefore the ciphertext is identical to the message!

   (a) Do you think the one-time pad scheme should be modified so that the all-zero key is not used? Explain.

   (b) Explain how it is possible that the one-time pad is perfectly secure even though the above situation can occur with non-zero probability.

   One to three sentences should be enough to answer each of these questions.

2. The following illustrate the magnitude of numbers encountered in cryptography.

   (a) DES is a block cipher which takes 64-bit input and produces 64-bit output; DES keys are 56 bits long. DES is sometimes considered to be a pseudorandom functions. How many bits are required to store a truly random function mapping 64-bit inputs to 64-bit outputs? Assume we wanted to store a random function from $\{0,1\}^n$ to $\{0,1\}^n$ on a 5-Gigabyte hard drive. What is the largest value of n we could support in this case?

   (b) AES is a block cipher with 128-bit keys. Assume we want to perform a brute-force search through the key-space of AES in order to break, say, an encryption scheme constructed using AES. Assume we use a 500 Mhz computer and assume (for simplicity) that we can test 1 AES key per clock cycle on this computer. How many AES keys can be tested per second on this computer? How long would it take (in years) to search through all 2^{128} possible keys on this computer?

3. Let F denote a block cipher which maps 64-bit inputs to 64-bit outputs, has 64-bit keys, and is assumed to be a (t, ϵ)-secure PRF for $t \approx 1$ year. What, if anything, can you say about the values of the following expressions:

   (a) $\Pr[k \leftarrow \{0,1\}^{64}$: the final bit of k is $0]$

   (b) $\Pr[k \leftarrow \{0,1\}^{64}$: the final bit of $F_k(0^{64})$ is $0]$

   (c) $\Pr[k \leftarrow \{0,1\}^{64} : x \leftarrow \{0,1\}^{64}$: the final bit of $F_k(x)$ is equal to the final bit of $x]$

   (d) $\Pr[k \leftarrow \{0,1\}^{64}; x \leftarrow \{0,1\}^{64}$: the final bit of $F_k(x) \oplus F_k(x)$ is $0]$

   (e) $\Pr[k \leftarrow \{0,1\}^{64}$: the final bit of $F_k(k)$ is $0]$

4. Consider the keyed function $G : \{0,1\}^{64} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ defined as follows: $G_s(x) = s \oplus x$. The goal of this problem is to show that G is not a very good PRF by constructing an efficient algorithm A for which:

$$\left| \Pr[f \leftarrow \mathsf{Rand}^{64 \rightarrow 64} : A^{f(\cdot)} = 1] - \Pr[s \leftarrow \{0,1\}^{64} : A^{G_s(\cdot)} = 1] \right| > 1/4. \tag{1}$$

   (a) Describe such an algorithm A. For the algorithm you specify, evaluate both $\Pr[f \leftarrow \mathsf{Rand}^{64 \rightarrow 64} : A^{f(\cdot)} = 1]$ and $\Pr[s \leftarrow \{0,1\}^{64} : A^{G_s(\cdot)} = 1]$. (The difference between these two values should be larger than $1/4$.)

   (b) Suggest a way to modify A so as to further increase the value of expression (1).

   (c) What is the best algorithm A you can come up with (i.e., the one maximizing expression (1)) subject to the restriction that it is only allowed to ask a *single* query to its "black-box"? Can you prove that no better algorithm is possible?

5. Let $P : \{0,1\}^k \times \{0,1\}^m \rightarrow \{0,1\}^m$ be a (t,ϵ)-secure PRP. Consider the encryption scheme defined as follows: the sender and receiver share in advance a randomly-chosen key $s \in \{0,1\}^k$. To encrypt a message $M \in \{0,1\}^{m/2}$, the sender chooses a random "padding" $r \in \{0,1\}^{m/2}$, concatenates r and M, and sends $C = P_s(r \circ M)$.

   (a) How can decryption be performed in the above scheme?

   (b) Consider the security of the above scheme against chosen-plaintext attacks (exactly as defined in class). Specifically, bound the success probability of any adversary A (running in time at most t) attacking the above scheme.

   (c) We can modify the above scheme to support encryption of m-bit messages in the following way: to encrypt an m-bit message M, simply break M in two parts M_1, M_2 and separately encrypt both halves. In class we gave the following encryption scheme for m-bit messages: $\langle r, P_s(r) \oplus M \rangle \leftarrow \mathcal{E}_s(M)$. Discuss the relative merits of these two encryption schemes for m-bit messages in terms of ciphertext length, security, and necessary conditions on P.

6. Consider the following MAC: Let $F : \{0,1\}^k \times \{0,1\}^m \rightarrow \{0,1\}^m$ be a (t,ϵ)-secure PRF for t very large and ϵ very small. The sender and receiver share a random secret key $sk \in \{0,1\}^k$ and fix ℓ which is assumed to be known to the adversary. Let $\langle i \rangle$ denote the ℓ-bit representation of integer i. To authenticate message M, the sender parses M as a sequence of $(m-\ell)$-bit blocks $M_1, \ldots, M_n$ (assume that message lengths are always a multiple of $(m-\ell)$, and that $n < 2^\ell$), chooses a random $r \in \{0,1\}^m$ and computes:

$$T = F_{sk}(r) \oplus F_{sk}(\langle 1 \rangle \circ M_1) \oplus \cdots \oplus F_{sk}(\langle t \rangle \circ M_n).$$

   The sender sends both T and r as the message authentication code for M.

   (a) How can the receiver verify correctness of a tag (T, r) on message M?

   (b) Suggest an attack on the above scheme which is better than brute-force search for the key sk, and describe the complexity of your attack and its probability of success in forging a valid tag on a new message. (I am aware of at least two different attacks, but any correct attack you give which is better than brute-force key search is fine.)