TD 2 – Block ciphers

Exercise 1. *False or false*

Explain why each of the following statements is wrong.

1. It is never possible to attack an ideal block cipher.
2. A block cipher with keys of 512 bits is always secure.
3. There will never be any reason, technologically speaking, to use (block cipher) keys larger than 128 bits.
4. One should always use (block cipher) keys larger than 128 bits.
5. One should always use the latest-published, most recent block cipher.

Exercise 2. *From the slides*

1. Prove that the four following informal security definitions for a block cipher E are encompassed by the (S)PRP security notion. *For each of them, assume you are given an efficient algorithm to break the security and build a adversary that has a large (S)PRP advantage.*
   i. Given $c = E(k,m)$, computing m without knowing k is *hard*.
   ii. Given m, computing $c = E(k,m)$ without knowing k is *hard*.
   iii. Given oracle access to E_k, it is *hard* to find k.
   iv. Given oracle access to $E_k^{\pm}$, it is *hard* to find k.
2. In the PRP experiment, assume that the challenger chooses $b \leftarrow \{0,1\}$ uniformly at random. Prove that for any adversary A, $\mathsf{Adv}_E^{\mathsf{PRP}}(A) = \left|2\Pr\left[\hat{b} = b\right] - 1\right|$, where $\hat{b}$ is the bit returned by the adversary.

Exercise 3. *Meet-in-the-middle and PRP advantage*

The meet-in-the-middle attack on double encryption allows an adversary to find the key from a pair (message, ciphertext), in time $O(2^\kappa)$ where κ is the length of each key. We translate this attack on a *lower* bound on the PRP advantage of double encryption.

Let $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ be a block cipher, where $\mathcal{K} = \{0,1\}^\kappa$ and $\mathcal{M} = \{0,1\}^n$. Let $EE_2 : \mathcal{K}^2 \times \mathcal{M} \to \mathcal{M}$ defined by $EE_2(k_1 \| k_2, m) = E(k_2, E(k_1, m))$.

1. Translate the meet-in-the-middle attack as an adversary A_{MITM} for the PRP experiment $\mathsf{Exp}_{EE_2}^{\mathsf{PRP}}$.
2. Give the number of queries to the oracle and the running time of A_{MITM}.
3. Give a lower bound on $\mathsf{Adv}_{EE_2}^{\mathsf{PRP}}(A_{\text{MITM}})$, and deduce a lower bound on $\mathsf{Adv}_{EE_2}^{\mathsf{PRP}}(q,t)$ for values q and t to be determined.

Time-memory trade-off. Consider the following variant of the meet-in-the-middle attack: Fix a length $\ell \le \kappa$; For all ℓ-bit strings $s \in \{0,1\}^\ell$, the adversary first computes (and stores) all the $y_{k_1} = E(k_1, m)$ for keys k_1 that begins with s and then test for each $k_2 \in \{0,1\}^\kappa$ whether $E^{-1}(k_2, c)$ belongs to the y_{k_1}'s; It stops if it finds a match, otherwise continues with the next prefix.

4. i. Analyze the time and space complexity of this attack.
   ii. Describe the attack in the two extremal cases $\ell = 0$ and $\ell = \kappa$.

Exercise 4. *Format-preserving encryption*

Consider a set $\mathcal{M}$ of message, distinct from $\{0,1\}^n$: say $\{0,1\}^{\le n}$ or the set of prime numbers $\le 2^{128}$, etc. A *format-preserving* block cipher is a block cipher for such an arbitrary set $\mathcal{M}$.

Assume that $\mathcal{M} \subset \{0,1\}^n$ for some n, and that we know an efficient algorithm that, given $m \in \{0,1\}^n$, determine whether $m \in \mathcal{M}$. The *cycle walking* algorithms convert a block cipher $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ to a format-preserving block cipher $E' : \{0,1\}^\kappa \times \mathcal{M} \to \mathcal{M}$. To encrypt $m \in \mathcal{M}$ using E' with a key k, compute $m' = E(k,m)$; If $m' \in \mathcal{M}$, return $c = m'$; Otherwise iterate with $m'' = E(k, m')$, etc.

1. Give the decryption algorithm $E'^{-1} : \{0,1\}^\kappa \times \mathcal{M} \to \mathcal{M}$.
2. Why is the existence of an efficient algorithm to test the appartenance to $\mathcal{M}$ not sufficient for E' to be efficient?
3. (*) Prove that the expected number of calls to E in the random oracle model is $(2^n + 1)/(|\mathcal{M}| + 1)$. *Hint. Prove (or admit) the following: given a size-t subset U of a size-N set S, the expected number of elements we need to sample (without replacement) from S to get an element of U is $(N+1)/(t+1)$.*