

---

## Problem Set 3 Solutions

**Problem 1. [60 points]** Let  $\mathcal{K}$  be the key-generation algorithm that returns a random 128-bit string as the key  $K$ . Let  $\mathcal{E}$  be the following encryption algorithm, based on the block cipher AES:

```
algorithm  $\mathcal{E}_K(M)$ 
  if  $|M| \neq 64$  then return  $\perp$  // Only encrypts 64 bit messages
   $R \xleftarrow{\$} \{0, 1\}^{64}$ 
   $C \leftarrow \text{AES}_K(R||M)$ 
  return  $C$ 
```

Above, “ $x||y$ ” denotes the concatenation of strings  $x$  and  $y$ , and  $x \xleftarrow{\$} \{0, 1\}^{64}$  is the operation of picking a random 64 bit string and calling it  $x$ .

1. [10 points] Specify a decryption algorithm  $\mathcal{D}$  such that  $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$  is a symmetric encryption scheme providing correct decryption.

We use the fact that AES is a block cipher and thus given the key one can easily compute its inverse  $\text{AES}^{-1}$ . The decryption algorithm is then as follows:

```
algorithm  $\mathcal{D}_K(C)$ 
  if  $|C| \neq 128$  then return  $\perp$ 
   $X \leftarrow \text{AES}_K^{-1}(C)$ 
  Break  $X$  into 64-bit blocks,  $X = R||M$ 
  return  $M$ 
```

Note that  $R$  is simply discarded; the decryption algorithm has no use for it.

2. [40 points] Give the best attack you can on this scheme. Your attack should take the form of an ind-cpa adversary  $A$  that makes  $q$  oracle queries and has  $O(q)$  running time, and you should specify  $\text{Adv}_{\mathcal{SE}}^{\text{ind-cpa}}(A)$  as a function of  $q$ . Indicate roughly for what value of  $q$  the advantage is at least  $1/2$ . (The better the attack, meaning the smaller the value of  $q$  for which this is true, the more points you get.)

Let  $q$  be an integer parameter,  $1 \leq q < 2^{64}$ . Our adversary works as follows:

```
Adversary  $A^{\mathcal{E}_K(\text{LR}(\cdot, b))}$ 
  for  $i = 1, \dots, q$  do
     $C_i \xleftarrow{\$} \mathcal{E}_K(\text{LR}(\langle i \rangle, 0^{64}, b))$ 
  if  $\exists i < j$  such that  $C_i = C_j$  then return 1
  else return 0
```

Above,  $\langle i \rangle$  denotes the representation of integer  $i$  as a binary string of length exactly 64 bits. For the analysis, let  $R_i$  denote the random choice made by the encryption algorithm in the computation of the response to the  $i$ -th LR-encryption query.

Let us first consider world 0, where  $C_i = \text{AES}_K(R_i \parallel \langle i \rangle)$  for all  $i = 1, \dots, q$ . Clearly  $\langle i \rangle \neq \langle j \rangle$  for  $i \neq j$ . This implies that  $R_i \parallel \langle i \rangle \neq R_j \parallel \langle j \rangle$  for all  $i \neq j$ , regardless of the values of  $R_i, R_j$ . Since  $\text{AES}_K$  is a permutation, this means that  $\text{AES}_K(R_i \parallel \langle i \rangle) \neq \text{AES}_K(R_j \parallel \langle j \rangle)$  for all  $i \neq j$ . In other words, in world 0,  $C_i \neq C_j$  for all  $i \neq j$ . This means that  $A$  always returns 0 in world 0. Thus

$$\Pr \left[ \mathbf{Exp}_{\mathcal{SE}}^{\text{ind-cpa-0}}(A) = 1 \right] = 0 .$$

Now let us consider world 1, where  $C_i = \text{AES}_K(R_i \parallel 0^{64})$  for all  $i = 1, \dots, q$ . Clearly  $R_i \parallel 0^{64} = R_j \parallel 0^{64}$  iff  $R_i = R_j$ . Thus,  $C_i = C_j$  iff  $R_i = R_j$ . In other words, the probability that  $A$  returns 1 is exactly the probability that there exist  $i < j$  such that  $R_i = R_j$ . This probability is exactly  $C(2^{64}, q)$ , the probability of a collision in throwing  $q$  balls into  $2^{64}$  bins. So we get

$$\Pr \left[ \mathbf{Exp}_{\mathcal{SE}}^{\text{ind-cpa-1}}(A) = 1 \right] = C(2^{64}, q) .$$

The advantage of  $A$  is thus

$$\begin{aligned} \mathbf{Adv}_{\mathcal{SE}}^{\text{ind-cpa}}(A) &= \Pr \left[ \mathbf{Exp}_{\mathcal{SE}}^{\text{ind-cpa-1}}(A) = 1 \right] - \Pr \left[ \mathbf{Exp}_{\mathcal{SE}}^{\text{ind-cpa-0}}(A) = 1 \right] \\ &= C(2^{64}, q) . \end{aligned}$$

We know that this is approximately equal to  $q^2/2^{65}$ . However, an approximation is not ideal here. What we would really like to do is lower bound the above. We can use Theorem A.1 of the Appendix on the birthday problem. It tells us that

$$C(2^{64}, q) \geq 1 - e^{-q(q-1)/2^{65}} .$$

We want to find the smallest value of  $q$  for which the above expression is at least  $1/2$ . Computation shows that the value of the expression is  $\approx 0.39$  when  $q = 2^{32}$  and  $\approx 0.86$  when  $q = 2^{33}$ , so  $q = 2^{33}$  works.

- 3. [10 points]** As a result of your attack, do you consider the scheme to be secure or insecure? Why?

Insecure. The attack provided above breaks the scheme using  $2^{33}$  queries, and computation of the same order. We are talking about encrypting  $2^{33} = 8,589,934,592$  or about 8 billion messages. This is not a large number in practice. Imagine a situation where packets are being encrypted on a fast (1Gbit/sec, say) network. It will take very little time before the number of messages encrypted is this number. But the attack says that you cannot encrypt this many message securely.

Of course there are other issues and arguments too. If we are encrypting email, you might say 8 billion messages is a large number and thus the attack is not practical. True, but that does not mean the scheme is secure. Remember as cryptographers we do not want to constrain the use of the scheme to certain applications. This is too complicated: no matter what, once a scheme is out there, people use it for everything. It better work even in extreme situations. Along the same lines you might argue that the particular attack is not important because it leaks very little information. Again, not a good viewpoint. We have seen many reasons to view IND-CPA as the right notion of security, so the attack is relevant.

