Cryptography

Programming 5

Ex 1 Implement a CPA experiment. Your program takes works in two modes:

encryption oracle – takes as input a set of messages m^i and returns corresponding ciphertexts c^i.

challenge – takes as an input two messages m_0, m_1, generates a random bit b and returns a corresponding ciphertext c_b.

Your program should support different modes of operations *i.e.,* AES_cbc_encrypt, use *openssl* library.

Ex 2 Implement a CPA distinguisher for the modified AES_cbc_encrypt. Assume that (the program implemented in 1) generates IV by incrementing its value by 1 each time a message is encrypted (compare with the Problem 3 of Assignment 3). You can achieve such a behaviour by providing appropriate value of $ivec$ to the (*include*/*openssl*/*aes.h*):

void *AES_cbc_encrypt*(const unsigned char $*in$, unsigned char $*out$,
size_t $length$, const AES_KEY $*key$,
unsigned char $*ivec$, const int enc);