Anish Laddha
Professor Avinash Kak
ECE 404
15 January 2024

HOMEWORK 01

Recovered Text

Charles Marc Herve Perceval Leclerc born 16 October 1997 is a Monegasque racing driver, currently racing in Formula One for Scuderia Ferrari. He won the GP3 Series championship in 2016 and the FIA Formula 2 Championship in 2017. Leclerc made his Formula One debut in 2018 for Sauber, a team affiliated with Ferrari, for which he was part of the Ferrari Driver Academy.

Recovered Key

Decimal: 1616
Hexadecimal: 0x0650

Math Explanation

Before explaining the code, I will explain the "math" behind the encryption and decryption. In encryption, there are 2 cases. The base case is for the 0^{th} block, where enc_0 = initial_vector ^ plain_0^key_vector. "_x" means the xth block of the file/string. The nth case is enc_n = enc_(n-1) ^ plain_n ^ key_vector. Concatenating all the enc_n's will give you the encrypted file.

For decryption, we use a few mathematical principles of XORing. First, a^(b^c) = (a^c)^b. Then, a^a = 0 and b^0 =b. This implies that, if given x = a^b, we can extract b if we know a and x. x^a = a^b^a = a^a^b = 0^b = b. now applying this to decryption, we can XOR our encrypted block with all of its component blocks except for the plain text in order to obtain the currently "unknown" plain text.In short, the base case is plain_0 = enc_0^initial_vector^key, and the nth case is plain_n = enc_n ^ enc_(n-1)^key.

Code Explanation

For lines 20-22, we load the file into a string, and then load that string of hex values into a bitvector.

For lines 30-34, we calculate the initial_vector by breaking the passphrase into blocks of size "bsize" (16 in our case), and then converting it into a bitvector and XORing them together

In lines 36-47, we implement the math I explained previously. First, we create an empty, decrypted bitvector. Then, we loop through the encrypted bitvector in chunk sizes of size "bsize." We store this encrypted text in a "prev" variable as we will need it in the next iteration. Then, to decrypt we XOR the encrypted with the previous encrypted chunk as well as the key vector to get the original plain text vector. We concatenate that to our decrypted vector. After we process the entire encrypted vector, we then convert the bitvector to text and return.