Jarod Nakamoto

CS 460

Assignment 3

1. The authors proposed rules 1 and 2 as a solution to characters being decrypted into garbage. Which is a problem that stems from the incorrect assumptions or practices of cryptographers.
    a. Rule 1: “Work with bytes not strings” addresses problems 1 through 3 in the article. Problems 1 and 3 stem directly from the assumption that characters are one byte of memory. However, characters can be represented in multiple bytes like in Unicode so this assumption causes problems when encrypting. If the cryptographic function uses the 1 character 1 byte assumption, then when its taking in a string then it will receive more characters than it should if the characters are multiple bytes. For example, a character using 2 bytes of space would be interpreted as 2 single byte characters and wrong ones at that. It makes sense that garbage would appear when trying to decrypt the cipher text because only half of the character information is there and there are twice as many characters (or more depending on how many bytes are per character). Problem 2, the sender, and the receiver have different character encoding, also makes decrypting difficult. The authors proposed the solution of using bytes and not strings to enable the encryption algorithm to account for these problems at a byte level instead of a character level which should let the program read the data properly.
    b. Rule 2: "Do not put cipher text bytes directly into a string type" addresses problem 4 of the article. The authors’ reasoning for this rule is that the cipher text bytes are a “random sequence of bytes with no structure and so decoding back to a string may not work” resulting in garbage and/or storing it in a string could alter the byte values. Additionally, they state that the output of encoding the decoded cipher text bytes does not always yield the same cipher text.
    c. Byte Order Mark is a Unicode character, U+FEFF, that lets big endian machines detect they are reading little endian data because it would be read as an illegal character. It is needed for sending data from little endian to big endian machines but not for UTF-8 data.
    d. The authors recommend storing/transmitting cipher text bytes in hexadecimal or in Base64 with a “"-----BEGIN FOO----" and "-----END FOO-----" encapsulation” or adding newlines.

1. (the second one)
    a. The ECB output is: 1111 0101 1111 0100 0000 0110 1001 0000. The problem with this output is that the key results in 0000 and the xor compliment results in 1111 and the result of 1111 is the xor compliment. Being only 4 bits of information there are 32 possible combinations for the key making it extremely easy to brute force decrypt.
    b. The CBC output is: 1000 0111 0010 1100 0110 1010 1001 0011 with a 4 Bit IV of 1101 (the counter, 0010, xor with result of the first ECB block,1111).
2. We need padding for encryption to make input the exact size needed for certain algorithms. The five methods of padding are same number, 0x80 and null (zeroes), zeroes and final counter byte, all null (zeroes), all spaces (0x20). Padding is not needed when plaintext is a multiple of block length and sender and receiver agree.