Basic Encryption Algorithms

In C++, there are two easy ways to implement encryption and both use the logical XOR (exclusive OR). All logical operators do bitwise operations on the individual bits in an integer data type. For example:

This example encrypts ‘s’ using ‘x’ as the encryption key.

Char ‘s’ (int value 115) 01110011

Char ‘x’ (int value 120) 01111000 XOR

00001011 = 11

These values are stored in an array of chars that represent the encrypted string. To decrypt the byte, you use the same encryption key. It does not take much to make this algorithm more complex: simple mathematical operations can serve to obfuscate the original identity of the to-be-encrypted number even more.

If the user passwords are stored in a database, most database managers have built-in encryption and decryption functions. These functions are inherently more complex and secure than the basic encryption algorithms discussed above, and would not be implemented on our side.

A possible encryption algorithm that is implementable is as follows:

1. Upon creation of a new user account, the program prompts the user to input a series of random letters for use as his/her encryption and decryption key. We can hash this key or encrypt it using a different method.

2. Next, whatever passwords the user enters will be encrypted using the aforementioned key and stored in a database. When the user seeks to retrieve a password, the key will be decrypted and then used in the decryption process for the password.