

Forum:[Hack Forums](#)

Thread: [\[C++\] BitCrypt Text Cipherer/Decipherer \[Version ALPHA 4\] \[ALPHA 5 IN-PROGRESS\]](#)

Sajuuk - September 8, 2012, 8:20 am

ALPHA 4 (Rev. 1): [\(Click to View\)Click Here For Current BitCrypt Pastebin Source Code...](#)

ALPHA 5 Will Include:

- RandChr() Function

Credits:

C++™ - Main Idea

Love2D - Fixed case problems. General help with C++.

void00 - General help with C++.

Daylloween - September 8, 2012, 8:29 am

Sounds pretty nice, what storage method are you using? Resources? If yes then i might know how you do it :)

Sajuuk - September 8, 2012, 8:32 am

Storage? What the heck is storage? Idk what that is.

notsteve - September 9, 2012, 2:27 pm

What is bitcrypt and what advantages does it offer over other options such as TrueCrypt?

e: assuming it's an encryption software because of your req. Idk what it is, is it a crypter?

Augustus - September 9, 2012, 2:41 pm

(Sep 8, 2012 08:32 AM)C++™ Wrote: [\[CITE\]](#) Storage? What the heck is storage? Idk what that is. The person is assuming your project is a crypter.

Sajuuk - September 9, 2012, 4:38 pm

Oh no. It's not a crypter. It's a text encryption program.

Love2D - September 9, 2012, 4:40 pm

Just text encryption? Are you kidding me?

Sajuuk - September 9, 2012, 4:52 pm

Hey how about you make a text encryptor? I am trying to learn how to do the actual things to encrypt it in the first place. I have this code from a website:

Spoiler [\(Click to View\)](#)Code #include <iostream>

#include <stdlib.h>

#include <cstdlib>

#include <windows.h>

#include <string.h>

using namespace std;

```

int main()
{
    string keyword;
    string msg;
    int crypted;
    int real_difference;
    int i=0;
    int k;
    int l;

    cout<<"Enter the keyword : ";
    getline(cin, keyword);

    cout<<"Enter the message you want to crypt : ";
    getline(cin, msg);

    k = keyword.length();
    l = msg.length();

    loop1:
    if(k < l); //used to even out the keyword and the sentence
    { //to change each letter
        keyword = keyword + keyword;
        k = keyword.length();
        if(k < l)
            goto loop1;
    }

    cout<<"\n\n";
    while (i<= msg.length()) //records length and uses string as array
    {
        keyword[i] -= 'a'- 1;

        if((msg[i] + keyword[i]) > 'z') //If crypted letter goes beyond 'z'
            crypted = 'a' + (keyword[i]+msg[i] - 'z') - 1;

        else if(msg[i] == 32) // If letter is a space
            crypted = msg[i];

        else
            crypted = msg[i] + keyword[i]; //Crypting

        cout<<(char)crypted; //Prints out the crypted letter
        i++;

        if(i>= msg.length())
            break;
    }

    return 0;

}

```

It's a Vigenere cipher code. How do I reverse the process?

Love2D - September 9, 2012, 5:42 pm

You are asking for help with this on cplusplus.com? You shouldn't be releasing it if you are going to have someone give you the code.

Sajuuk - September 9, 2012, 6:24 pm

Hello! They will be credited with creating it. A full credits list will be implemented in the program. Duh.

Love2D - September 9, 2012, 6:39 pm

But if encryption and decryption is the only thing the program does then you are asking for the whole program to be done for you!

Sajuuk - September 9, 2012, 6:55 pm

Did I ask for the whole program? I want to have an example to refer back to when I'm stuck, which I am currently. All I want is to figure out how to encrypt text.

Love2D - September 9, 2012, 7:01 pm

You already copied the Vigenere Cipher from another website. Are you sure you are just using this code to refer back to when stuck?

Sajuuk - September 10, 2012, 1:33 pm

As you can see, if you look, the old Batch BitCrypt was based off of AMDjack's HXB-128 cipherer. It used IFs to read the plaintext, convert it to another specified character, then return the value and go back to the loop with GOTO. In this case now, I'll be using IF, FOR, WHILE or GOTO, etc. If only I had a caesar cipher code to refer to...

Sylex - September 10, 2012, 5:27 pm

Why go through all the work when it's already been done? Just be sure to credit the author of this neat lib.

http://embeddedsw.net/libObfuscate_Cryptography_Home.html

Just be sure to implement a correct mode of operation. Read up on them here:

http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

I personally suggest GCM, but replace the incremental aspect of it with libOfuscate's CSPRNG, and seed it with part of the expanded password.

For a password expansion algorithm I recommend scrypt:

<http://www.tarsnap.com/scrypt.html>

scrypt has a variety of unique features which makes it extremely secure for a password expansion hash, such-as all the features of bcrypt but with variable memory usage, making it much harder to create a hardware implementation of the algorithm.

Also please for the love of all cryptology, don't start creating new algorithms unless you want some extremely weak software.

Sajuuk - September 10, 2012, 5:32 pm

@Sandstone

I want to make my own text cipherer/decipherer. I need a quality code to look back on when I need to learn. Still need a sample code though :'(

Sylex - September 10, 2012, 5:36 pm

(Sep 10, 2012 05:32 PM)C++™ Wrote: [\[CITE\]](#) @Sandstone

I want to make my own text cipherer/decipherer. I need a quality code to look back on when I need to learn. Still need a sample code though :'(

Links to source code are on some of the pages I sent you. Specifically the first and last links.

Sajuuk - September 10, 2012, 5:45 pm

By my own I mean coded by me, not including others' files. I will include others' code, but will not link files. I know this sounds weird, but the Batch BitCrypt was standalone, and I want it to be like that when in C++. UPDATED OP.

Tenacious - September 10, 2012, 6:44 pm

Why would you use the Caesar Cipher, it's pretty much the most simple encryption algorithm.

You should just write your own algorithms if you are just doing a text Encryptor.

Sajuuk - September 10, 2012, 6:57 pm

All I need is some code to look back on. I actually need to learn how to cipher a letter to begin with. Anyone have a sample code on how to cipher an individual letter?

Love2D - September 10, 2012, 7:00 pm

If you don't even know the basics of string manipulation then why are you attempting to make something like this?

Sajuuk - September 10, 2012, 7:07 pm

Isn't C++ easier to code string manipulation than Batch? I thought I could count on you all to help me. I guess I was wrong :'(.

Love2D - September 10, 2012, 7:13 pm

It really depends on what you are doing.

I would answer your specific questions at cplusplus.com, but I don't do much here.

Tenacious - September 10, 2012, 7:27 pm

(Sep 10, 2012 06:57 PM)C++™ Wrote: [\[CITE\]](#) All I need is some code to look back on. I actually need to learn how to cipher a letter to begin with. Anyone have a sample code on how to cipher an individual letter?

You don't need to use any ciphers that exist, it would actually be less secure if you did.

String manipulation can be really easy.

Anyways, I wrote the ceasar cipher for you, you can add operations to the encryption function to make it more secure and unique.

```
Code /*
```

```
Demonstrated the Caesar Cipher Encryption Algorithm
```

```
Written by Illuminexus
```

```
*/
```

```
#include <iostream>
```

```

#include <windows.h>
#include <string.h>

using namespace std;

int i = 0;
int shifter = 3;
string text = "";

void encryptText()
{
    cout << "Your Normal Text is: " << text << "." << endl;

    for (i = 0; i < text.length(); i++)
    {
        text [i] = (text [i] + shifter);
    }

    cout << "Your Encrypted Text is: " << text << endl;
}

void getText()
{
    cout << "Please Enter some Text: ";
    getline(cin,text);
    cout << "Please enter the character Shift Amount: ";
    cin >> shifter;
    cout << endl;
    cout << endl;
    encryptText();
}

int main()
{
    getText();
    system("pause");
}

```

Love2D - September 10, 2012, 7:30 pm

(Sep 10, 2012 07:27 PM)iluminexus Wrote: [\[CITE\]](#) You don't need to use any ciphers that exist, it would actually be less secure if you did.

String manipulation can be really easy.

Anyways, I wrote the ceasar cipher for you, you can add operations to the encryption function to make it more secure and unique.

```

Code /*
Demonstrated the Caesar Cipher Encryption Algorithm
Written by Illuminexus
*/

```

```

#include <iostream>
#include <windows.h>
#include <string.h>

using namespace std;

int i = 0;
int maxLetters = 26;
int shifter = 3;
string text = "";

void encryptText()
{
    cout << "Your Normal Text is: " << text << "." << endl;

    for (i = 0; i < text.length(); i++)
    {
        text [i] = (text [i] + shifter);
    }

    cout << "Your Encrypted Text is: " << text << endl;
}

void getText()
{
    cout << "Please Enter some Text: ";
    getline(cin,text);
    cout << "Please enter the character Shift Amount: ";
    cin >> shifter;
    cout << endl;
    cout << endl;
    encryptText();
}

int main()
{
    getText();
    system("pause");
}

```

There is really no need for global variables here. Not to mention the system call AND variables that should be constants.

Code for (i = 0; i < text.length(); i++)

This is calling a function(err, member function) for every character in the string. Q.Q

Tenacious - September 10, 2012, 7:40 pm

(Sep 10, 2012 07:30 PM)Love2D Wrote: [CITE](#) There is really no need for global variables here. Not to mention the system call AND variables that should be constants.

Code for (i = 0; i < text.length(); i++)

This is calling a function(err, member function) for every character in the string. Q.Q

I know it's not perfect, I did it up real quick, and don't insult my global variables, I love them, even

though they're not necessary lol ;)

I don't expect him to copy and paste, so he'll need to write his own variation.

iVision - September 11, 2012, 1:38 am

I've seen you wanted a cipher method, (didn't read the rest replies.) Its just this:

```
Code #include <stdio.h>
#include <string.h>

int main()
{
    int nCiph = 14;
    char szHello[] = "Hello World!";
    char* szBuff = malloc(strlen(szHello) * sizeof(char));

    int i;
    for (i = 0; i < strlen(szHello); i++)
        szBuff[i] = szHello[i] + nCiph;

    printf("%s", szBuff);
    free(szBuff);

    return 0;
}
```

Not sure if it compiles but I think you get the idea.

Just PM, or reply if you need some more help with this, and I'll help you out ;)

Grrr...Trolls - September 11, 2012, 9:36 am

Here is some code you could use....

```
Code if program->coded == false then
{
    program->codeNow(bitcrypt);
    program->code = true;
}
else
{
    return 1; //fail
}
```

Sajuuk - September 11, 2012, 1:30 pm

DashRantic@stackoverflow.com posted this code to generate random chars:

```
Code #include <iostream>
#include <string.h>
#include <cstdlib>

using namespace std;

static const char alphanum[] =
"0123456789"
```

```
"!@#$%^&*"
"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
"abcdefghijklmnopqrstuvwxyz";
```

```
int stringLength = sizeof(alphanum) - 1;
```

```
char genRandom()
{
    return alphanum[rand() % stringLength];
}
```

```
int main()
{
    while(true)
    {
        cout << genRandom();
    }
    return 0;
}
```

Now this generates random chars. I am going to implement this function into BitCrypt. It will be used for the functions RandKey and RandString. These functions work as follows:

RandKey:

Generates a random key. Can be turned on or off.

RandString:

Generates random length (1-300) of random chars and adds them to random places in the resulting ciphertext. The strings are stored in a separate file (when one string ends, "\n" or "endl" is used to start a new line), and when one wants to decipher a message ciphered with BitCrypt, they will have to insert the ciphered message and place the output "RandChar.txt" in the same location as "BitCrypt.exe". BitCrypt will then remove the random chars listed in "RandChar.txt" from the ciphertext, and reverse the ciphering process to decipher the message. An example would be:

Result CipherText Would Look Something Like This [\(Click to View\)](#)Code
h043fu04fu0843u3joswolej434tu3toi3jrjiojfe9023ljfijrooij4gi4j

RandChar.txt [\(Click to View\)](#)Code 043fu04fu0843u3joswol
j434tu3toi3jrjiojfe9023
jfijr
oij4gi4j

PlainText [\(Click to View\)](#)Code Hello

tflo - September 11, 2012, 2:07 pm

```
Code char randChr()
{
    return rand()%('~' - ' ' + 1) + ' ';
}
```

```
string randKey(int len)
```

```

{
    string out;
    while (len-- > 0)
        out.push_back(randChr());
    return out;
}

```

Irritable - September 11, 2012, 2:11 pm

Oh my gosh. This entire thread has made me laugh so much. Thanks!

The best text encryption method is the XOR method. Just refer to the Windows OS as proof!

Sajuuk - September 11, 2012, 2:17 pm

You laugh at those who are attempting to program their first REAL program? Sorry I even posted this then. So...I have this code now:

```

Code #include <windows.h>
#include <string.h>
#include <iostream>
#include <ctime>
#include <cstdlib>

using namespace std;

int main()

{
    string input;
    cout << "Enter plaintext: ";
    getline(cin, input);
}

void randnum()
{
    srand((unsigned)time(0));
    int random_integer;
    for(int index=0; index<1; index++){
        random_integer = (rand()%300)+1;
        cout << random_integer << endl;
    }
}

void randchar()
{
    static const char alphanum[] =
        "0123456789"
        "!@#$%^&*"
        "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        "abcdefghijklmnopqrstuvwxyz";

    int stringLength = sizeof(alphanum) - 1;

    char genRandom()

```

```

    {
return alphanum[rand() % stringLength];
    }

    int main()
    {
while(true)
    {
        cout << genRandom();
    }
    }
}

void caesar1()
{
    for (ctr = input.length;ctr >= input.length;ctr++)
    {
        if (in[ctr]=='a')
        {
            out[ctr] = 'z';
        }
    }
}

```

Is there a better way to change to the next char in the PlainText than using a FOR loop? Or is that the only way?

Love2D - September 11, 2012, 3:35 pm

(Sep 11, 2012 02:17 PM)C++™ Wrote: [\[CITE\]](#) You laugh at those who are attempting to program their first REAL program? Sorry I even posted this then. So...I have this code now:

```

Code #include <windows.h>
#include <string.h>
#include <iostream>
#include <ctime>
#include <cstdlib>

```

```

using namespace std;

```

```

int main()

{
    string input;
    cout << "Enter plaintext: ";
    getline(cin, input);
}

```

```

void randnum()
{
    srand((unsigned)time(0));
    int random_integer;
    for(int index=0; index<1; index++){
        random_integer = (rand()%300)+1;
        cout << random_integer << endl;
    }
}

```

```

    }
}

void randchar()
{
    static const char alphanum[] =
        "0123456789"
        "!@#$%^&*~"
        "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        "abcdefghijklmnopqrstuvwxyz";

    int stringLength = sizeof(alphanum) - 1;

    char genRandom()
    {
        return alphanum[rand() % stringLength];
    }

    int main()
    {
        while(true)
        {
            cout << genRandom();
        }
    }
}

```

```

void caesar1()
{
    for (ctr = input.length; ctr >= input.length; ctr++)
    {
        if (in[ctr]=='a')
        {
            out[ctr] = 'z';
        }
    }
}

```

Is there a better way to change to the next char in the PlainText than using a FOR loop? Or is that the only way?

Why the fuck would you have this function?

```

Code void randnum()
{
    srand((unsigned)time(0));
    int random_integer;
    for(int index=0; index<1; index++){
        random_integer = (rand()%300)+1;
        cout << random_integer << endl;
    }
}

```

Sajuuk - September 11, 2012, 3:45 pm

That generates the number of chars to input into the specified string. I actually need another one to generate the number of random strings. Thanks.

captmicro - September 11, 2012, 3:48 pm

Stop posting shitty ass ciphers that can be broke with a few minutes and a debugger. Anything that isn't standardized is vulnerable to cryptanalysis and most likely very weak. If you knew anything about crypto you would know that. You should use a real encryption algorithm like DES (triple-DES is better as DES is vulnerable to some attacks) or Blowfish that has been tested many millions of times for weaknesses, not some home-brew cipher that took 5 seconds to write. Also most AV's can scan for XOR'd strings (considering the key can only be 0-0xFF for single-character (but you could XOR 4-byte blocks or something)) by just XORing said string with all possible keys. Even a rolling key XOR would be better than your standard for (`; *x; x++) *x ^= XOR_KEY;`;

Love2D - September 11, 2012, 3:51 pm

Why do you have a for loop that only loops once? Why do you have a variable to hold the random number? So many questions.

Your code is just TERRIBLE, and I'm trying to be as nice as possible here.

Quote: Stop posting shitty ass ciphers that can be broke with a few minutes and a debugger. Anything that isn't standardized is vulnerable to cryptanalysis and most likely very weak. If you knew anything about crypto you would know that. You should use a real encryption algorithm like DES (triple-DES is better as DES is vulnerable to some attacks) or Blowfish that has been tested many millions of times for weaknesses, not some home-brew cipher that took 5 seconds to write. Also most AV's can scan for XOR'd strings (considering the key can only be 0-0xFF for single-character (but you could XOR 4-byte blocks or something)) by just XORing said string with all possible keys. Even a rolling key XOR would be better than your standard for (`; *x; x++) *x ^= XOR_KEY;`;

This.

Sajuuk - September 11, 2012, 3:57 pm

Well excuse me for trying to learn.

captmicro - September 11, 2012, 5:01 pm

Here is a [RC6](#) cipher that will hopefully show you how a real cipher works.

<https://dl.dropbox.com/u/3657276/crypto.zip>

NOTE: this has never been compiled / used, just an implementation I made from the rc6 example source code (google for it) that uses a structure for key data (really not needed, but this is part of a crypto lib I'm currently working on so the structs help keep it clean if I need more than just a key for a cipher)

Sajuuk - September 11, 2012, 6:26 pm

@captmicro

Code doesn't compile.

```
Code::Blocks Error Log (Click to View) Code C:\Users\admin\Desktop\C++\RC6\rc6.h
|4|error: typedefs.h: No such file or directory|
C:\Users\admin\Desktop\C++\RC6\rc6.h
|26|error: expected initializer before 'void'|
|=== Build finished: 2 errors, 0 warnings ===|
```

I wish I had bookmarked that code I saw earlier...

Xori - September 11, 2012, 6:42 pm

Quote: All I need is some code to look back on. I actually need to learn how to cipher a letter to begin with. Anyone have a sample code on how to cipher an individual letter?

Could you please display some brain activity, or even creativity?

Let's start with something simple.

HurrDurrCipher:

Each letter in your input has a position in the alphabet. Load up the alphabet in reverse into an array, and to "encrypt" your text, use the indices in the legitimate alphabet on the reverse alphabet.

e.g index of 'a' in the regular alphabet is 0.

The letter at [0] in your reverse alphabet is Z. So, aaa -> zzz. Congrats, shit custom cipher.

Quote:If you knew anything about crypto you would know that. You should use a real encryption algorithm like DES (triple-DES is better as DES is vulnerable to some attacks) or Blowfish that has been tested many millions of times for weaknesses, not some home-brew cipher that took 5 seconds to write.

The guy is making one of his first programs, it isn't a commercial application. Also in case (apparently) you missed it, he's not making a crypter to conceal things from AVs, he's dealing with text files. I'm assuming he intends it to run on "home-brew ciphers that took 5 seconds to write" (maybe better ones as he progresses), if not, yeah, you can just go and grab some industrial algorithms...

You can pretty much do anything you want, this isn't a commercial application and you won't actually produce anything secure without doing some legitimate studying. Just try all kinds of fun tricks etc.

captmicro - September 11, 2012, 7:26 pm

(Sep 11, 2012 06:26 PM)C++™ Wrote: [CITE](#) @captmicro

Code doesn't compile.

```
Code::Blocks Error Log (Click to View)Code C:\Users\admin\Desktop\C++\RC6\rc6.h
[4]error: typedefs.h: No such file or directory|
C:\Users\admin\Desktop\C++\RC6\rc6.h
[26]error: expected initializer before 'void'|
||=== Build finished: 2 errors, 0 warnings ===|
```

I wish I had bookmarked that code I saw earlier...

Ha, forgot to include the typedef header from it, here ya go:

```
Code #ifndef _typedefs_h_
#define _typedefs_h_
```

```
typedef char CHAR; typedef short SHORT; typedef int INT;
typedef unsigned char UINT8, UCHAR, BYTE, *PCHAR;
typedef unsigned short UINT16, USHORT, WORD;
typedef unsigned int UINT32, UINT, ULONG, DWORD, BOOL;
```

```
#define TRUE 1
#define FALSE 0
```

```
#endif // _typedefs_h_
```

edit: removed some code from it you don't need

Sajuuk - September 12, 2012, 5:28 pm

(Sep 11, 2012 06:42 PM)Xori Wrote: [\[CITE\]](#) The guy is making one of his first programs, it isn't a commercial application. Also in case (apparently) you missed it, he's not making a crypter to conceal things from AVs, he's dealing with text files. I'm assuming he intends it to run on "home-brew ciphers that took 5 seconds to write" (maybe better ones as he progresses)

Thanks for realizing this. You're the only one who did :P. So, I've come up with this:

```
Code // Test Code ONLY
// Not A Commercial Program OR A Crypter
// THIS IS A TEXT CIPHERER

#include <windows.h>
#include <cstdlib>
#include <iostream>

using namespace std;

int main()
{
    string in;
    string out;
    char lower [] = { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z };
    char upper [] = { A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z };
    char lowercip [] = { z, y, x, w, v, u, t, s, r, q, p, o, n, m, l, k, j, i, h, g, f, e, d, c, b, a };
    char uppercip [] = { Z, Y, X, W, V, U, T, S, R, Q, P, O, N, M, L, K, J, I, H, G, F, E, D, C, B, A };
    cout << "Enter PlainText: ";
    getline(cin, in);
    caesar();
}

void caesar()
{
    // what goes here? This question is the Xori's post. Can anyone help?
}

@Xori
```

Exactly how do I "use the indices in the legitimate alphabet on the reverse alphabet"? Does this mean to retrieve the next char from the input variable? With the current code, it gives errors:

Code 'a' was not declared in this scope

I've included <vector> and <array>, but it says <array> is experimental. Is this safe? Also, how do I split a string into a char array (not an int or string array)?

captmicro - September 12, 2012, 6:44 pm

Wikipedia has some real nice diagrams on Caesar ciphers if you want to check it out

http://en.wikipedia.org/wiki/Caesar_cipher.

And by "use the indices in the legitimate alphabet on the reverse alphabet" I think he means use the index you used on the normal alphabet array as the index you use on the reversed one, such as (you only need the reversed arrays for this, not the normal ones):

```
Code char c = 0;
```

```
char *out = HeapAlloc(GetProcessHeap(), HEAP_ZERO_MEMORY, strlenA(in));
for (; *in; ++in) //loop till *in == 0
{
    if ((*in >= 'a') || (*in <= 'z')) //lowercase
    {
        *out = lowerReversed['z' - *in]; out++;
    }
    else if ((*in >= 'A') || (*in <= 'Z')) //uppercase
    {
        *out = upperReversed['Z' - *in]; out++;
    }
}
HeapFree(GetProcessHeap(), 0, out);
At least I think that is what you want (I really doubt this for some reason as this is basically strrev).
For real caesar (although this IS a type of circular caesar shift) shifts look at the wiki page above.
```

lucky13 - September 12, 2012, 7:40 pm

(Sep 11, 2012 03:48 PM)captmicro Wrote: [\[CITE\]](#) Stop posting shitty ass ciphers that can be broke with a few minutes and a debugger. Anything that isn't standardized is vulnerable to cryptanalysis and most likely very weak. If you knew anything about crypto you would know that. You should use a real encryption algorithm like DES (triple-DES is better as DES is vulnerable to some attacks) or Blowfish that has been tested many millions of times for weaknesses, not some home-brew cipher that took 5 seconds to write. Also most AV's can scan for XOR'd strings (considering the key can only be 0-0xFF for single-character (but you could XOR 4-byte blocks or something)) by just XORing said string with all possible keys. Even a rolling key XOR would be better than your standard for (; *x; x++) *x ^= XOR_KEY;

If you'r gonna try real encryptions algorithms why not just go with AES? Blowfish is still alright but DES is pretty much broken and AES in ECB mode isn't too complicated. Maybe something like sosemanuk if you want to look at stream ciphers. As for all your caesar ciphers you could just use an array with the alphbet in it, a switch to find your letters index and use modulo 26 to find the index of the letter your substituting... Eff I'll edit that later with some code so it makes sense.

captmicro - September 12, 2012, 8:07 pm

Personally I just use rc6 since you can implement it in like 50 lines of code, and example C source is available online.

Sajuuk - September 13, 2012, 4:34 am

Why must I use already made crypts? I'm making a text cipherer, not a crypter for FUDing.

Cybride - September 13, 2012, 5:02 am

It is insecure.

[Video: http://www.youtube.com/watch?v=3Re5xlEjC8w&feature=player_embedded]

Sajuuk - September 13, 2012, 1:08 pm

@Cybride

I'm not making a commercial crypter, I'm making a personal text cipherer for fun. Please understand that I'm not making something to FUD a program.

void00 - September 13, 2012, 1:46 pm

Code I'm not making a commercial crypter, I'm making a personal text cipherer for fun. Please understand that I'm not making something to FUD a program.

If it is for personal use and like as you mentioned, for fun, I'd encourage you to develop an

encryption algorithm on your own,
while learning about cryptography.

If you plan to release your app, it'd be better to use algorithms which have been proven strong by cryptanalysis.

These algorithms were tested for a long period of time, for weaknesses and only the best are chosen as standard.

This will ensure that your application stands up to the cryptography standards.

Sajuuk - September 13, 2012, 1:49 pm

Oh I have a question for you all. I know that multidimensional arrays take up a lot of space and all, but can I use it for a simple vigenere cipher, or is there a better way to make something like this?:

WARNING: Large Image From Wikipedia ([Click to View](#))

[img]http://upload.wikimedia.org/wikipedia/commons/9/9a/Vigen%C3%A8re_square_shading.svg[/img]

iVision - September 13, 2012, 2:09 pm

Why do you need array? For selecting random chars or smething? Just use rand() in the right ascii character space (safes memory resource)

void00 - September 13, 2012, 2:11 pm

well, in the wikipedia page, you can see the algebraic description of the cipher

encryption is done as:

<http://upload.wikimedia.org/math/a/8/6/a8693a20966b33fbdf5967cf253085e2.png> (the image's not displaying properly on dark grey background)

Code $C_i = E_k(M_i) = (M_i + K_i) \bmod 26$ where C_i is the cipher character, M_i is the plaintext character at position i ,

K_i is the i th character of key

First, you need to convert your plaintext and key to upper case.

If the key is not as long as the plaintext, append the key with itself till the length matches.

then use the above function to encrypt it.

For decryption, its just the reverse.

No need to use a multi-dimensional array

Sajuuk - September 15, 2012, 5:56 am

I came up with this code:

```
Code #include <windows.h>
#include <iostream>
#include <cstdlib>
#include <string>
#include <cmath>
```

```
using namespace std;
```

```
int main()
```

```

{
    int ctr;
    int ctr1; // input counter
    int ctr2; // cipher counter
    string output; // output string to display. Arrays for ciphering are as follows:
    char cipalp [26] = {'z','y','x','w','v','u','t','s','r','q','p','o','n','m','l','k','j','i','h','g','f','e','d','c','b','a'};
    char regalp [26] = {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
    string input; // input string initialized
    getline(cin, input); // Get user input.
    ctr = input.length() + 1; //it includes the null at the end
    char *arr = new char[ctr]; // make a new array (dynamic size)
    for(int i=0; i < ctr; i++) // FOR loop to create array from input variable.
    {
        arr[i] = input[i];
    }
    ctr2 = 0;
    for (ctr1 = 0; input.length() < ctr1; ctr1++) {
        ctr2++;
        if (arr[ctr1] == regalp[ctr2]) {
            output = output + cipalp[ctr2];
        }
    }
    cout << output;
    delete arr;
    return 0;
}

```

For some reason, it doesn't display the resulting CipherText. Is the FOR loop correct? It compiled fine with no errors or warnings, but it just doesn't want to show me the ciphertext.

EDIT: It freezes when I remove the line: Code `ctr = input.length() + 1;`. Why is this?

void00 - September 15, 2012, 10:53 am

Ok, I'm assuming that you are trying to implement the basic substitution cipher here.

So, a will be replaced by z, b by y, ...etc rt.

First, to make ur string to an char array, you can use something like the following instead of the for loop.

for loops works too btw.

Code `char *arr = new char[ctr];`

`strcpy(arr, input.c_str());`

Secondly, i think you got the condition in your second for loop wrong. Shouldn't it be:

Code `ctr1 < input.length()` Also, instead of using `input.length()`, you can first set `ctr=ctr-1` and then use `ctr` instead of `input.length()`

That's just a suggestion though.

Inside the for loop, the condition inside the if is an assignment. You have to use `==` operator and not `=`.

Also, you're only checking if `arr[ctr1] == regalp[ctr2]` , if it is not, then if is skipped.

I think you need one more for loop, something like this :

Code `ctr=ctr-1;`

for (`ctr1 = 0; ctr1 < ctr; ctr1++`) {

 for(`ctr2=0;ctr2<26;ctr2++`){

```
    if (arr[ctr1] == regalp[ctr2]) {  
        output = output + cipalp[ctr2];  
    }  
}  
}
```

Hoping that helps.

Sajuuk - September 15, 2012, 10:23 pm

Big thanks void00. See OP for updated Pastebin source code.

void00 - September 16, 2012, 2:24 am

I think you forgot to put

Code `ctr=input.length()+1`;before you create a new char array `arr`

You're also decrementing `ctr` at 2 places (line 24 and 32). I don't know if you're doing that on purpose though.

Also, if i think your code works without some of the includes,

You might just be needing `iostream` and `cstring`

Do implement decryption soon. And prompt the user for input.

Make it user friendly.

Sajuuk - October 14, 2012, 3:45 pm

Revision of how BC will work:

1. String is inserted using `getline(cin, var)`;
2. String is parsed char by char and the ciphered char is added to the output string.
3. When loop is done, it returns the variable.

Does anyone know if this will work?