

GoGCryptLib encryption library

Part 1: Description

GoGCryptLib is the custom GusOnGames encryption library. We are a small company developing custom software and mainly games. The encryption algorithm used in the library was developed entirely in our labs. We have used techniques originally found in other industry standard algorithms and we combined them in a way we believe is quite strong and trustworthy. The main advantage of the algorithm is that it is not documented which makes it even harder to break.

It was initially created more than twenty years ago and it is constantly improved ever since. We feel quite confident with the robustness of the algorithm, so that we decided to develop a product based on it.

Part 2: What it can do

The encryption library provides two basic functions. One to encrypt and one to decrypt a buffer. This is actually what anyone needs from an encryption library. The sample application shows you how to use these functions to encrypt or decrypt a file.

The internal parameters of the functions are different from the ones we used in our free file encryption utility so that files encrypted with one cannot be decrypted by the other, so you can be sure that there is no free tool out there that can decrypt your data.

Part 3: How it is done

The information provided from this point on is rather technical written with the developer in mind.

As we said in the part 2 the library exports two main functions. These are

```
int GOGCRYPTLIB_API gogEncryptBuff(unsigned char* buff, long size, const char* pass);  
int GOGCRYPTLIB_API gogDecryptBuff(unsigned char* buff, long size, const char* pass);
```

It is quite obvious what they do. The first encrypts and the second decrypts a memory buffer. The arguments they expect are identical. The first is the buffer you want to encrypt/decrypt, the second is the number of bytes you want to modify (this must not exceed the actual length of the buffer) and finally the password to be used in the encryption/decryption process. Both functions return 1 for success and 0 for failure.

Finally the library has an error reporting function

```
int GOGCRYPTLIB_API gogGetLastError();
```

This function can be used to see what went wrong. The only thing we think might go wrong is common programming errors like NULL pointer input buffer or password or zero as length of input buffer. To be honest those were the only things that went wrong in our implementations over the years.

Part 4: Package description

The package you download contains two libraries. The first is the actual **GoGCryptLib** and the second is the **GoGCryptDev** library. The **GoGCryptLib** is encrypted. You can use the **GoGCryptDev** which uses the password to XOR the bytes in the buffer until you fully develop your application and then purchase a license and a password to unlock the **GoGCryptLib** library. The tool to perform this action is provided in the package.

Part 5: Installation

The installation of the developer's toolkit is very simple. The whole package is only two (2) libraries, two header files and two DLLs. Since we are aiming to developers we thought it would be simpler to describe a manual setup operation along with the automatic setup utility.

Installation procedure number 1: Download the automatic installer and proceed with the installation. When the installation has finished you should make your development tool point to the 'include' directory for compiler resolution of the include files and, to the appropriate library directory (x64 or win32) for linker input resolution. Automatic installation generates a link to the library decryption tool.

Installation procedure number 2: Download the 'zip' compressed archive and copy its contents to a folder of your choice. Pointing to the include and library folders applies to this procedure as well. In the case of manual installation you have to create the link to the decryption tool yourself.

Part 6: Decrypting the library

As we said before the actual encryption library is encrypted as well. In order to get its decryption password you have to purchase a license. Visit the site and register (if you haven't done already). Then you can access the 'Buy Now' button. After you finish the purchase process you will receive an email with the unlock password. Type it in the bottom line of the decryption window, browse and select the correct DLL and then press the decrypt button. The same decryption code applies for both the 64 and the 32 bit DLLs. There is no limit in the use of the decrypted DLL in any way. Once decrypted it can be copied to any computer as explained in the next section. Just make sure that you have issued the SAME email address in both the registration form and in PayPal and that this is the email you have typed in the unlock utility. The unlock code you receive is connected with this email address and any mistypes will generate errors.

Part 7: What do I get with the license?

Once you have the decryption code and you decrypt the library you can use it in as many programs as you like. You do not have to get a new license for every product you develop. As simple as that. We do not like complex legal documents describing software licenses, you buy the library and it is yours to use in any product you like. There is no limit in the number of developers in your company that will be using it either.

Part 8: Summary...

This covers all the bases I think. You have seen what we offer, how to obtain a license and what it covers, and of course how to install the library and develop your products using it. If you feel like you want some more information or help about any related subject please send us an e-mail at crypthelp@gusongames.com. We will be happy to offer any help.

Gus