Subject: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Wed, 16 Sep 2009 20:17:54 GMT
View Forum Message <> Reply to Message

Sometimes we may have task to store some large file (4+ GB) or small string inside encrypted storage. I tried to make a pair classes which make it easy. This is the first version, so any ideas are welcome.

This package assumes you have OpenSSL library successfully installed and its paths are added to TheIDE.

OK, let me introduce a pair of classes called AESEncoderStream and AESDecoderStream. They support streamed adding and encryption/decryption of data. Encryption is made with AES (Rijndael) with 128, 192 or 256 bit keys.

Encrypted data 32 bytes larger than source length aligned to 16-byte boundary. I.e. if your source data is 170 bytes long, the resulting length is:
170 rounded by 16-byte pieces = 176
plus
32 (header data)
= 176 + 32 = 208 bytes.
Not so ugly for a number of applications especially if source data is large.

Here is a simple self-explanating demo:

```
#include <Core/Core.h>
#include <openssl/aes.h>
#include <AESStream/AESStream.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 AESInit();

 // Generate cryptographically stable key
 String key(AESRandomString(32));

 // Encryption
 String sIn,sOut;
 sIn =
"qwertyuiop[p\tasdfghjkl;zxcvbnm,./quwiueqiwueoiquweioquweioquweiqwueicuwinuqiweqiwue pqi
ueci eiqniuriryuweyruweyruewrycuwbrurbywuyrwquiercbbcrebrquwey";
 AESEncoderStream aesEncoder(sIn.GetLength(), key);
 aesEncoder << sIn.Left(10);
 aesEncoder << sIn.Mid(10,10);
 aesEncoder << sIn.Right(sIn.GetLength() - 20);

 sOut << aesEncoder; //do streamed encoding
```

```
 // Decryption
 //key.Set(0, 'a'); //uncomment to see what happens with wrong key
 AESDecoderStream aesDecoder(key);

 aesDecoder << sOut.Left(15); //you may add by parts
 aesDecoder << sOut.Right(sOut.GetLength() - 15);

 try
 {
  String sDecoded;
  sDecoded << aesDecoder; //throw exception if key is wrong

  Cout() << (sDecoded == sIn) << "\n\n"; //check if all converted successfully
 }
 catch (const char *xp)
 {
  Cout() << "\n!!Error: " << ToSystemCharset(xp);
 }
}
```

File Attachments

1) AESStream.zip, downloaded 838 times

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Thu, 17 Sep 2009 07:34:16 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

How are you installed OpenSSL in Windows ?

I have seen an installer in here http://www.slproweb.com/products/Win32OpenSSL.html but I would like to know your opinion.

Best regards
Koldo

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Weras on Thu, 17 Sep 2009 17:57:43 GMT
View Forum Message <> Reply to Message

Hi koldo!

I had resolved this problem and I did so.

1. Download openssl-0.9.8k.tar.gz from
 https://www.openssl.org/source/
2. Unpack archive to C:\temp\openssl-0.9.8g
3. Download and install ActivePerl
4. Now type in the command line:
 1) C:\temp\openssl-0.9.8g>perl Configure VC-WIN32 --prefix=c:/temp/openssl-bin/
 2) C:\temp\openssl-0.9.8g>%comspec% /k ""c:\Program Files\Microsoft Visual Studio 8\VC\vcvarsall.bat"" x86
 3) C:\temp\openssl-0.9.8g>ms\do_masm.bat
 4) if you need static library write
 C:\temp\openssl-0.9.8g>nmake -f ms\nt.mak
 else, if you need dynamic library write
 C:\temp\openssl-0.9.8g>nmake -f ms\ntdll.mak
5. The result is files*.lib & *.dll and include directory
6. Add .../openssl/inc32 as Include and .../openssl as Linker directories

Is enough to work with openssl

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Thu, 17 Sep 2009 20:55:10 GMT
View Forum Message <> Reply to Message

Hello Weras and Mindtraveller

Just excellent.

The Weras instructions worked perfectly the first time .
Sorry to say that it is unusual.

And Mindtraveller wrapper and demo worked well the first time .

Function dword rdtsc() uncludes a little of assembler so it only compiles with MSC. I will try to translate it to Gcc to see if it works with MinGW and Linux.

Best regards
Koldo

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Fri, 18 Sep 2009 09:20:49 GMT
View Forum Message <> Reply to Message

Hi, koldo!

It was Weras who instructed me how to install OpenSSL so I asked him to answer yur question.

Thanks for trying Linux version, it's a great effort.
I have just found a pair of small bugs and reuploaded sources. If you wish to try AESStream, pleas rewrite sources with new versions. If you manage to make POSIX version of rdtsc - it would be great too (I'm no professional in GCC-accepted assembler).

I wonder if people need this too. If it is so, someone from authors may upload AESStream to official Bazaar.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by tojocky on Fri, 18 Sep 2009 10:28:40 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Fri, 18 September 2009 12:20Hi, koldo!

It was Weras who instructed me how to install OpenSSL so I asked him to answer yur question.

Thanks for trying Linux version, it's a great effort.
I have just found a pair of small bugs and reuploaded sources. If you wish to try AESStream, pleas rewrite sources with new versions. If you manage to make POSIX version of rdtsc - it would be great too (I'm no professional in GCC-accepted assembler).

I wonder if people need this too. If it is so, someone from authors may upload AESStream to official Bazaar.

Very nice package and wiki how to build openssl on win32.
Thank you!

Ion Lupascu (tojocky)

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Fri, 18 Sep 2009 11:23:27 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

About rdtsc, in fact it compiles in gcc as it has an ifdef so that:

- If MSC, it takes a value from rdtsc (time stamp counter 64-bit register)
- Else it takes Random()

It seems that to get a random number our Random() implementation is better than the clock (MT19937 algorithm), so perhaps rdtsc() would have to be changed with Random()

I have tried to compile in MinGW, but I get linking errors, in summary:

Openssl\out32\libeay32.lib(tmp32/ui_openssl.obj),(.text[_rea d_string_inner]+0xb): undefined reference to `__security_cookie'
Openssl\out32\libeay32.lib(tmp32/ui_openssl.obj),(.text[_rea d_string_inner]+0x149): undefined reference to `@__security_check_cookie@4'
Openssl\out32\libeay32.lib(tmp32/ecp_smpl.obj),(.text[_ec_GF p_simple_group_set_curve]+0x6): undefined reference to `_chkstk'

Does anybody know how to solve these problems with chkstk and security_cookie ?

Best regards
Koldo

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Fri, 18 Sep 2009 12:01:30 GMT
View Forum Message <> Reply to Message

Yes, I tried it too and had the same problems.
Currently had no time to discover this (actually MSC9 satisfies me on Win32), but some googling could help.

UPDATE: this link might be useful
http://wagner.pp.ru/~vitus/articles/openssl-mingw.html

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Fri, 18 Sep 2009 21:01:54 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

Thank you for your reference.

Unfortunately it has been impossible for me to compile it directly, with MSys or with Cygwin.
In any case sooner or later I get an error that finish the process.

Best regards
Koldo

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by kasome on Wed, 23 Sep 2009 01:29:13 GMT
View Forum Message <> Reply to Message

Great job.

Thanks, Mindtraveller.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Sat, 20 Feb 2010 16:08:36 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

AESStream is not included in Bazaar yet.

I think it would be good to include it

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Sat, 20 Feb 2010 22:19:44 GMT
View Forum Message <> Reply to Message

Please upload it. It looks like I have no access to bazaar.

Koldo, please tell, did you try to compile this under POSIX? Did it work well or you had any problems. It would be very good to keep this solution really cross-platform (at least MSC9/Win32 + GCC/POSIX).

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Sun, 21 Feb 2010 07:07:49 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

GCC/POSIX works perfect.

MSC9/Win32 works well for me only in non DEBUG mode.

When I run WIN32 version in DEBUG mode I get a "Heap leaks detected". I get the same using the openssl .dll version.

I think the problems come from Web/SSL package, not from AESStream.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Sun, 21 Feb 2010 09:38:41 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

I have got also a "Memory leaks detected" compiling uppdev/textssl with MSC in debug mode.

Its code is just:

#include <Core/Core.h>

CONSOLE_APP_MAIN
{

}

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Sun, 21 Feb 2010 11:01:41 GMT
View Forum Message <> Reply to Message

Koldo, what is the package uppdev/textssl? Never heard of that before. You mean another OpenSSL-based package?

I also changed package a bit, added tutorial (English and Russian versions), added example. Please be so kind check if it compiles (currently don't have OpenSSL installed) and upload it to bazaar.

File Attachments
1) AESStream.zip, downloaded 660 times

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Sun, 21 Feb 2010 14:40:40 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Sun, 21 February 2010 12:01Koldo, what is the package uppdev/textssl? Never heard of that before. You mean another OpenSSL-based package?

I also changed package a bit, added tutorial (English and Russian versions), added example. Please be so kind check if it compiles (currently don't have OpenSSL installed) and upload it to bazaar.
Hello Mindtraveller

Web/SSL package is included in AESStream. A problem is it have memory leaks. I have found a solution to solve it, that is in Web/SSL/util.cpp:

INITBLOCK {
 Socket::Init();
 CRYPTO_set_mem_functions(SSLAlloc, SSLRealloc, SSLFree);
 SSL_load_error_strings();

```
 //SSL_library_init();  //SOLUTION TO MEMORY LEAK !!
}
```

Probably SSL_library_init(); is not necessary and as it not properly cleaned up, there are memory leaks.

To find this problem I have used uppdev/testssl package and Upp technology to detect memory leaks

I have tested your changes but they do not work well . Check this:

sIn includes this:
Quote:qwertyuiop[p asdfghjkl;zxcvbnm,./quwiueqiwueoiquweioquweioquweiqwueicuwin uqiweqiwue pqiueci eiqniuriryuweyruweyruewrycuwbrurbywuyrwquiercbbcrebrquwey

sDecoded iincludes this:
Quote:qwertyuiop[p asdfghjkl;zxcvbnm,./quwiueqiwueoiquweioquweioquweiqwueicuwin uqiweqiwue pqiueci eiqniuriryuweyruweyruewrycuwbrurbywuyrwquiercbbcrebr

---

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Sun, 21 Feb 2010 15:01:12 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

MinGW works well using openssl .dll version.

Just add this to Web/SSL in "Package organizer":

File Attachments

1) sc.PNG, downloaded 2237 times

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Sun, 21 Feb 2010 18:25:59 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

Other question. In the demo program you generate a key with AESRandomString(32).

I have integrated successfully AESStream in a program (very easy, just using your demo), but

including in the program an array with the key previously generated with AESRandomString().

Is it possible to use instead an user generated key (a password) ?

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Sun, 21 Feb 2010 23:48:53 GMT
View Forum Message <> Reply to Message

Hello, Koldo.

I have found bug in AESStream which led to different original and decoded strings in the example. Also I got rid of Web/SSL dependency. Just tested it, and everything seems fine.

Please try updated version of AESStream and read it's tutorial if you want to use it in your app. Tutorial say that you should not use user password instead of generated key.

Why?
1) Key must be 128/192/256 bits long. User password may have ANY length.
2) Key is very important part of cryptographic strength of overall encryption. Using criptographically weak key (user password in 99,9% of cases is extremely weak) turns all AES encryption into weak and rather breakable system. As well as using cryptographically strong key makes overall AESStream system extremely unbreakable to anyone.

File Attachments

1) AESStream.zip, downloaded 691 times

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Mon, 22 Feb 2010 06:54:00 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

Quote:1) Key must be 128/192/256 bits long. User password may have ANY length.
2) Key is very important part of cryptographic strength of overall encryption. Using criptographically weak key (user password in 99,9% of cases is extremely weak) turns all AES encryption into weak and rather breakable system. As well as using cryptographically strong key makes overall AESStream system extremely unbreakable to anyone.

Does it mean that AES cannot be used for saving user files with user defined password ?

However there are programs that include this possibility with AES. For example 7zip offers AES-256 encryption http://www.7-zip.org/7z.html.

Is there a standard way to convert a 8 chars user defined password into an useful 256 AES bits key ?

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Mon, 22 Feb 2010 07:31:18 GMT
View Forum Message <> Reply to Message

koldo wrote on Mon, 22 February 2010 09:541) Does it mean that AES cannot be used for saving user files with user defined password ?

However there are programs that include this possibility with AES. For example 7zip offers AES-256 encryption http://www.7-zip.org/7z.html.

2) Is there a standard way to convert a 8 chars user defined password into an useful 256 AES bits key ?
1) Cryptography is no miracle, it's just math. If you use weak password, you get weak protection, and no algorithm saves you from it. This means if you want stable and strong protection, you must use stable and strong key. The one of few options here is to use key generated by OpenSSL itself.

You have to consider user password as worst type of key. Also, many passwords are too plain and dumb: 123, 111, 123456, etc. This is bad for cryptography.

Russian programmer Igor Pavlov who wrote 7zip, has chosen to use compromise solution. He takes user password, calculates SHA-256 function for it (AFAIK U++ has its realization too). Then he adds some calculations/changes to that 256-bit value and the final value is used as a key for AES encryption.

This represents fair protection, which is very much stronger than using user password as key, but at some rate weaker protection than with OpenSSL-generated key. In a number of uses it is rather good and satisfactory protection. Also it allows using protection without storing user password itself which is very good practice. But frankly speaking I haven't heard of SHA output as extremely cryptographically strong combination of bytes. This algorithm has another application field (generating unique digest "far" from original bytes).

2) AFAIK there is no "standard" way to convert user password to key. The best way is to use OpenSSL generated key. You may of course use any function like SHA-256 but you must be aware of the crytpographic strongness/weakness you give to user.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Mon, 22 Feb 2010 07:50:10 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Mon, 22 February 2010 08:31koldo wrote on Mon, 22 February 2010 09:541) Does it mean that AES cannot be used for saving user files with user defined password ?

However there are programs that include this possibility with AES. For example 7zip offers AES-256 encryption http://www.7-zip.org/7z.html.

2) Is there a standard way to convert a 8 chars user defined password into an useful 256 AES bits

key ?
1) Cryptography is no miracle, it's just math. If you use weak password, you get weak protection, and no algorithm saves you from it. This means if you want stable and strong protection, you must use stable and strong key. The one of few options here is to use key generated by OpenSSL itself.

You have to consider user password as worst type of key. Also, many passwords are too plain and dumb: 123, 111, 123456, etc. This is bad for cryptography.

Russian programmer Igor Pavlov who wrote 7zip, has chosen to use compromise solution. He takes user password, calculates SHA-256 function for it (AFAIK U++ has its realization too). Then he adds some calculations/changes to that 256-bit value and the final value is used as a key for AES encryption.

This represents fair protection, which is very much stronger than using user password as key, but at some rate weaker protection than with OpenSSL-generated key. In a number of uses it is rather good and satisfactory protection. Also it allows using protection without storing user password itself which is very good practice. But frankly speaking I haven't heard of SHA output as extremely cryptographically strong combination of bytes. This algorithm has another application field (generating unique digest "far" from original bytes).

2) AFAIK there is no "standard" way to convert user password to key. The best way is to use OpenSSL generated key. You may of course use any function like SHA-256 but you must be aware of the crytpographic strongness/weakness you give to user.

Excellent explanation

I will follow your advice. Anyway, could you add a function to convert an username password into a "fair" protection ?. Thanks

I have checked your demo and now it works well. In a big program where I have applied it, it works well too .

You have done more changes than just a fix . You have removed dependencies to packages Web and Web/SSL.

This afternoon I will upload it to Bazaar. In some hours I will propose a possible application of your useful functions.

Great job !

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Mon, 22 Feb 2010 10:55:00 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

I have compiled OpenSSL in MinGW, although only obtaining .dll . It is very easy.

If you want I can include it in T++.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Mon, 22 Feb 2010 11:24:51 GMT
View Forum Message <> Reply to Message

Yes, it would be great to add article "Building OpenSSL with MINGW under Windows" and add its reference into tutorial article.
Great work, Koldo, and thanks for checking out my AESStreams!

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Mon, 22 Feb 2010 11:46:55 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

Your package is very useful.

If somebody requires to encrypt seriously a String or raw data from small size to Gb, this is a simple way to do it .

This is not an encryption algorithms catalog. This is just one of the best options with an easy interface.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by tojocky on Mon, 22 Feb 2010 14:19:08 GMT
View Forum Message <> Reply to Message

Koldo,

Can you ad in BAZAAR?

Than you Koldo, and Mindtraveller!

Regards, ion Lupascu (tojocky).

koldo wrote on Mon, 22 February 2010 13:46Hello Mindtraveller

Your package is very useful.

If somebody requires to encrypt seriously a String or raw data from small size to Gb, this is a simple way to do it .

This is not an encryption algorithms catalog. This is just one of the best options with an easy interface.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Mon, 22 Feb 2010 15:46:35 GMT
View Forum Message <> Reply to Message

tojocky wrote on Mon, 22 February 2010 15:19Koldo,

Can you ad in BAZAAR?

Than you Koldo, and Mindtraveller!

Regards, ion Lupascu (tojocky).

koldo wrote on Mon, 22 February 2010 13:46Hello Mindtraveller

Your package is very useful.

If somebody requires to encrypt seriously a String or raw data from small size to Gb, this is a simple way to do it .

This is not an encryption algorithms catalog. This is just one of the best options with an easy interface.

Quote:This afternoon I will upload it to Bazaar.
Yes. In a few UTC hours

Subject: Building with MSC9
Posted by kohait00 on Wed, 03 Mar 2010 22:17:58 GMT
View Forum Message <> Reply to Message

hi people, great work...

compile with MSC9 worked great, both static and dynamic.
2 quick corrections / hints, though

1) using MSC9 stuff the command is

C:\Temp\openssl-0.9.8m> %comspec% /k ""c:\Program Files\Microsoft Visual Studio 9.0\VC\bin\vcvars32.bat"" x86

2) the out32/ folder offset for the *.lib stuff should be removed in package organizer of AESStreamTest, since the build method setup for LIB stuff already points to "C:\Temp\openssl-0.9.8m\out32"

Subject: Re: Building with MSC9
Posted by koldo on Sat, 06 Mar 2010 00:49:39 GMT
View Forum Message <> Reply to Message

Hello Kohait00

Quote:C:\Temp\openssl-0.9.8m> %comspec% /k ""c:\Program Files\Microsoft Visual Studio 9.0\VC\bin\vcvars32.bat"" x86
Included

Quote:the out32/ folder offset for the *.lib stuff should be removed in package organizer of AESStreamTest, since the build method setup for LIB stuff already points to "C:\Temp\openssl-0.9.8m\out32"

Sorry, I do not understand.

Subject: Re: Building with MSC9
Posted by kohait00 on Sun, 07 Mar 2010 10:24:27 GMT
View Forum Message <> Reply to Message

hi koldo,

i meant the following:

according to the instructions provided, the LIB directories are to be set to "C:\Temp\openssl-0.9.8m\out32", but in package organizer, the AESStream package includes "out32/libeay32.lib" or "out32dll/libeay32.lib" depending on flags.. this wont work, the libs wont be found.

the LIB directories entry should be set to "C:\Temp\openssl-0.9.8m", thats all.. sorry for misleading post.

Subject: Re: Building with MSC9
Posted by koldo on Sun, 07 Mar 2010 12:47:41 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Sun, 07 March 2010 11:24hi koldo,

i meant the following:

according to the instructions provided, the LIB directories are to be set to "C:\Temp\openssl-0.9.8m\out32", but in package organizer, the AESStream package includes "out32/libeay32.lib" or "out32dll/libeay32.lib" depending on flags.. this wont work, the libs wont be found.

the LIB directories entry should be set to "C:\Temp\openssl-0.9.8m", thats all.. sorry for misleading post.
Hello Kohait00

Now in AESStream doc it appears in the setup the next description:

Quote:...

2.5. The result is in next folders:
inc32: Include files
out32: *.lib files for static linking
out32dll: *.lib & *.dll files for dynamic linking

...

2.6. Add in "Setup/Build methods/Lib directories" menu, the directory where out32 and out32dll have been copied.

Is it ok ?

And sorry. Your post was not misleading. Just hard to understand for me

---

Subject: Re: Building with MSC9
Posted by kohait00 on Sun, 07 Mar 2010 15:08:05 GMT
View Forum Message <> Reply to Message

thnak you, no problem.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Wed, 10 Mar 2010 16:09:09 GMT
View Forum Message <> Reply to Message

Hello Mindtraveller

Could you add in AESStream the possibility of accepting keys with length different that 32, 16, ..., perhaps using the SHA-256 algorithm as you suggested before ?

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by kohait00 on Wed, 10 Mar 2010 19:54:30 GMT
View Forum Message <> Reply to Message

hi koldo,

as far as i got the point of mindtraveler, AES and the other symetric algorithms are not to be thought of beeing based on a "password", a user defined and therefore week combination of signs (which would be scanned first in a brute force attack), but on a statistically well distributed *binary* key (128 bit should be made wise . it is hard for a human beeing to generate one. so the computer will take over and provide some random ones(AES key generator). this key should be thought of as a "password", what it of corse isn't. everything else would diminish the stability of the key. maybe to get over it, think of it as kind a GUID which you generate once for your application (which in real world communication does not apply . dont think of AES as sort of alphanumerical password dependant encryption algorithm, it's indeed, just as mindtraveler mentioned: math. i had the luck to enjoy some lectures cryptology, and it confuses sometimes. but the first thing we learned there was to forget the idea of passwords / human readable strings as security base.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Wed, 10 Mar 2010 20:33:18 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Wed, 10 March 2010 20:54hi koldo,

as far as i got the point of mindtraveler, AES and the other symetric algorithms are not to be thought of beeing based on a "password", a user defined and therefore week combination of signs (which would be scanned first in a brute force attack), but on a statistically well distributed *binary* key (128 bit should be made wise . it is hard for a human beeing to generate one. so the computer will take over and provide some random ones(AES key generator). this key should be thought of as a "password", what it of corse isn't. everything else would diminish the stability of the key. maybe to get over it, think of it as kind a GUID which you generate once for your application (which in real world communication does not apply . dont think of AES as sort of alphanumerical password dependant encryption algorithm, it's indeed, just as mindtraveler mentioned: math. i had the luck to enjoy some lectures cryptology, and it confuses sometimes. but the first thing we learned there was to forget the idea of passwords / human readable strings as security base.
Yes yes, all of you are right

However think about for example a file encrypting software to be used by different people. How would you do it ?

Option 1: The software gives the user a 32 bytes random key
Option 2: The user enters a key

Option 1 seems much stronger. However file and hard disk encrypting software seems to choose option 2.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by kohait00 on Wed, 10 Mar 2010 21:36:12 GMT
View Forum Message <> Reply to Message

http://www.winzip.com/aes_info.htm
should explain that its not trivial

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Thu, 11 Mar 2010 08:12:20 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Wed, 10 March 2010 22:36http://www.winzip.com/aes_info.htm
should explain that its not trivial
Hello Kohait00

Thank you for the reference. I will use it.

Coming to the issue, look at this:

- If it is open source, I cannot put the key in the code

- If the program creates a key for the user, and he/she is not let to change it, a 32 bytes password seems too hard to use

- If we use a user defined key, we could include in AESStream:

---1. A SHA 256 possibility to convert user password in a 32 bytes key
---2. The means to avoid a brute force attack.
For example, if AES 256 with a weak user key can resist within and acceptable probability, for example, 1000000 random keys, AESStream could let the main program to enter, for example, 1000 keys per day and after that, AESStream would refuse any additional key.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by kohait00 on Thu, 11 Mar 2010 09:29:39 GMT
View Forum Message <> Reply to Message

i dont know if i remember it correctly, but there are several technices combined to achieve encryption of data trggered by a user password.

1) the en/de cryption is done using a *fast* (symetrical) algorithm, like AES (they are blockorientated and relatively similar, only differ in their block functions (F functions, or Feistel Function)

2) the key used there, is the key we were speaking about, and is encrypted and stored with the data. as encryption can be used slow but really strong asymetrical (public / private key) algorithms like RSA.

3) the password thing comes into play with things like diffie hellman secure exchage of information with having it travel over the net.

but its quite a while now, and i may mix it up with things like vpn tunneling and handshaking and so on..

but in any way: encrypting decrypting to fit current standards is far from beeing trivial and involves a lot of steps, password is only a small part of it, maybe we should stick to common technologie here (means in openssl)

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Thu, 11 Mar 2010 09:53:04 GMT
View Forum Message <> Reply to Message

The truth is that you MUST
1) Use strong key with AES, not password
2) Not to hardcode the key in the code in ANY way

So what is the solution? You take user password. And then you DERIVE strong key from it. Then you "forget" user password, you just don't need it at all. You do encryption with that relatively strong key (i.e. SHA from user password - see my recent comment).

Next time user enters password, you derive the key with the same function (i.e. SHA) and try to decompress AESStream. If decomression fails, then original password and the one entered is not the same (incorrect password).

It is really not that hard.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Thu, 11 Mar 2010 10:01:43 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 11 March 2010 10:53The truth is that you MUST
1) Use strong key with AES, not password
2) Not to hardcode the key in the code in ANY way

So what is the solution? You take user password. And then you DERIVE strong key from it. Then you "forget" user password, you just don't need it at all. You do encryption with that relatively strong key (i.e. SHA from user password - see my recent comment).

Next time user enters password, you derive the key with the same function (i.e. SHA) and try to decompress AESStream. If decomression fails, then original password and the one entered is not the same (incorrect password).

It is really not that hard.
Could you implement this in AESStream ?

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Mindtraveller on Fri, 12 Mar 2010 20:41:01 GMT
View Forum Message <> Reply to Message

It will take some time before I switch to AES again. But I will.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by koldo on Sun, 14 Mar 2010 14:27:33 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Fri, 12 March 2010 21:41It will take some time before I switch to AES again. But I will.

Subject: bazaar: DeEncrypter
Posted by kohait00 on Thu, 05 Aug 2010 19:09:25 GMT
View Forum Message <> Reply to Message

in bazaar is a DeEncrypter based on AESStream. i hope i am using it right (Mindtraveller could review it). it works so far, but only for files < 200 MB or so (allocator breaks with Out of Memory sometime, if file too big, it's the upp allocator i am using, default case).

so the matter is, it can import keys. and could be extended with a functionality to derive the key from a user password...

but best would be to have a port for RSA asymetrical encryption. then one could encrypt the key and leave it encrypted with the public key, stay with the data somewhere, and 'unlock' (decrypt) it with the password, which would result somehow in the private key.

as far as i remember the major encryption programs do it the same , maybe more fancy

Subject: Re: bazaar: DeEncrypter
Posted by Mindtraveller on Sat, 07 Aug 2010 05:14:36 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Thu, 05 August 2010 23:09in bazaar is a DeEncrypter based on AESStream. i hope i am using it right (Mindtraveller could review it). it works so far, but only for files < 200 MB or so (allocator breaks with Out of Memory sometime, if file too big, it's the upp allocator i am using, default case). Which package/file is to look at?

Subject: Re: bazaar: DeEncrypter
Posted by kohait00 on Sun, 08 Aug 2010 09:05:49 GMT
View Forum Message <> Reply to Message

the package is named 'DeEncrypter', there is no ref in uppweb yet, it's too young

Subject: Re: bazaar: DeEncrypter
Posted by Mindtraveller on Tue, 10 Aug 2010 13:33:19 GMT
View Forum Message <> Reply to Message

If I'm not mistaken, you load each file into the memory and then encrypt/decrypt it. If it's right, then you do wrong way.

The main idea behind AESStream classes is streaming. This means you don't need to load the file into memory. All you need is to open file and read its contents by small chunks into AESStream class then taking encrypted(decrypted) chunks from it into other file. You don't have to load file into memory. You don't have to handle large String object within your code.

I think the fact you have exception for large files means you should switch to chunks and streaming and refuse loading whole file into memory.

Subject: Re: bazaar: DeEncrypter
Posted by kohait00 on Tue, 10 Aug 2010 13:46:18 GMT
View Forum Message <> Reply to Message

yes, absolutely.

this was a quick shot, i needed it for my app, delivering some additional ressource files, which should be 'scrambled' for small files it works with no issues, i just wanted to try it with larger files ofcorse.

i am not very familiar with the Stream stuff in upp yet, but definitely need to..

meanwhile, i dont mind anyone to change the behaviour accordingly, this could, infact, be a true life example of how to do streaming processing with AESStream..if you already know what to change, feel free to do so i might do it as soon as i have some more info on that and some more time.. like always.

cheers

Subject: Re: bazaar: DeEncrypter
Posted by Mindtraveller on Tue, 10 Aug 2010 19:14:29 GMT
View Forum Message <> Reply to Message

Please read AESStream package help page. It has example of "right" approach to streaming. If you have any further questions please feel free to ask.

P.S. koldo, I remember I've promised to embed key generation functionality and I will do it in the near future. Excuse my delay.

Subject: Re: bazaar: DeEncrypter
Posted by kohait00 on Tue, 10 Aug 2010 19:42:26 GMT
View Forum Message <> Reply to Message

i've looked there before.but couldnt quite get well with the example.. after all, you too are using Strings sIn and sOut to contain loaded and generated data. i suppose i need to use FileStream at that point, and call flush from time to time...

Subject: Re: bazaar: DeEncrypter
Posted by Mindtraveller on Wed, 11 Aug 2010 07:43:44 GMT
View Forum Message <> Reply to Message

I've finally embedded user password hashing into AESStream classes. This means you may use password as argument without any potential security problems. AESStream classes calculate SHA256 hash from your password and use it as a key.

koldo, could you please take this archive and update packages in bazaar? There is also a new version of MtAlt there. Thanks in forward.

File Attachments
1) bazaar.zip, downloaded 519 times

Subject: Re: bazaar: DeEncrypter
Posted by koldo on Wed, 11 Aug 2010 13:29:58 GMT
View Forum Message <> Reply to Message

Excellent.

Now plain user password can be used!

Two things:

- As class has changed, old encrypted strings and keys do not match . Could you add the user password possibility not by default? (as it was before)

- In AESHashedString() you could just use SHA256String() function in Sha.cpp file .

Subject: Re: bazaar: DeEncrypter
Posted by Mindtraveller on Thu, 12 Aug 2010 08:22:43 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 11 August 2010 17:29- As class has changed, old encrypted strings and keys do not match . Could you add the user password possibility not by default? (as it was before)
- In AESHashedString() you could just use SHA256String() function in Sha.cpp file .
1) Yes. I'll make new classes with new names for updated behaviour, while old ones will behave like before.
2) No, I can't. SHA256String returns hex-formatted string, not hash itself.

Subject: Re: bazaar: DeEncrypter
Posted by koldo on Thu, 12 Aug 2010 11:54:49 GMT
View Forum Message <> Reply to Message

Perfect!

Subject: Re: bazaar: DeEncrypter
Posted by kohait00 on Tue, 17 Aug 2010 13:58:52 GMT
View Forum Message <> Reply to Message

DeEncrypter is now streaming the files properly (i hope).
if you could get a look into it...
the user password thing could be encorporated there as an extra button.. for loading the SHA256 key.. but i'm not yet familiar with this.

Subject: Re: bazaar: DeEncrypter
Posted by koldo on Fri, 25 Feb 2011 17:52:26 GMT
View Forum Message <> Reply to Message

Hello All

Now, as it should be from the beginning, SHA256String() and rest of SHA2 functions return the SHA hash in binary.

For all SHA2 functions there is a version that return the hash in Hex, like SHA256Hex().

Subject: Re: bazaar: DeEncrypter
Posted by Mindtraveller on Thu, 21 Apr 2011 12:28:06 GMT
View Forum Message <> Reply to Message

Currently I'm using AESStream heavily and must admit that automatic SHA hash from password provided is not a must. In a number of situation the original password is still needed.

Subject: Re: Encrypted storage with streaming (OpenSSL, AES)
Posted by Alboni on Thu, 22 Aug 2013 23:59:31 GMT
View Forum Message <> Reply to Message

Hello, is AESStream compatible with php ?
http://www.php.net/manual/en/function.mdecrypt-generic.php

I can't decrypt my messages using
http://aes.online-domain-tools.com/

(AES with 32 byte key in CBC mode)

Also I'm quite confused that there is no AES_decrypt() call anywhere in AESDecoderStream?