Subject: Cypher package - An extensible Encryption package
Posted by mdelfede on Sat, 02 Oct 2010 23:42:41 GMT
View Forum Message <> Reply to Message

I uploaded the very first implementation of a generic encryption package, as discussed in topic

http://www.ultimatepp.org/forum/index.php?t=msg&th=5568& amp; amp; amp;start=0&

I tried to have an interface as modular as possible in order to be able to merge all current Encryption packages.

It has a base class defining the interface; supports String, Block and Streaming encryption.
By now it implements RC4 and Snow2 Streaming symmetric encryptors.

Other modules should derive from CypherBase class and implement ALL of its pure virtual functions in order to keep the interface identical.

Package still miss error handling, it'll implemented when we'll agree on the interface proposed.
Docs are missing too, for the same reason.

There's also an extensible testing application, which allow to select the encryption module, the encryption mode and some more.

Pavel, could you look if the interface meets your needs too ?
It still miss the Initialization Vector handling, I've got some ideas on how to make it fit yours and my needs, but before implementing it I'd like the interface to be stable enough.

Ciao

Max

EDIT : Please wait to review package... I'm making still many changes in interface and moving most routines to base class.

Max

Subject: Re: Cypher package - An extensible Encryption package
Posted by mdelfede on Sun, 03 Oct 2010 18:38:46 GMT
View Forum Message <> Reply to Message

Interface should be stable enough, also the streaming part is working.

There are 2 functions to define in derived classes :

```
// main encoding/decoding function
```

```
        // must be redefined on each derived class
        virtual void Cypher(byte const *sourceBuf,  byte *destBuf, size_t bufLen) = 0;

        // main key setting function
        // must be redefined on each derived class
        virtual bool CypherKey(byte const *keyBuf, size_t keyLen, byte const *nonce, size_t
nonceLen) = 0;
```

Plus, you must give BlockSize in constructor in case of Block Cyphers like AES.

Streaming is done with << operator (stream in) and >> operator (stream out).
For block cyphers, there's a Flush() function which pads last block with random data and returns size of encoded stream (true size, without padding). When decoding, using SetStreamSize(size) allows the decoder to un-pad the last block and return cleaned stream.

Block mode is fully supported, with checking of block size in case of Block-Cyphers.
Encoding/Decoding in block mode is done by some overloaded operator() which supports String encoding and binary buffer encoding, in place and out of place.

The test app CypherTest now supports both test with block and streaming modes.

Still missing a couple of small stuffs, but it should be stable enough now.

Ciao

Max

Subject: Re: Cypher package - An extensible Encryption package
Posted by Mindtraveller on Tue, 05 Oct 2010 07:00:00 GMT
View Forum Message <> Reply to Message

Thanks! I'll look at it today and write ASAP.

Subject: Re: Cypher package - An extensible Encryption package
Posted by mdelfede on Tue, 05 Oct 2010 12:20:05 GMT
View Forum Message <> Reply to Message

I guess I need to add an option to have embedded in encoded stream both the IV and the data size.... Not too difficult indeed.
I'll wait for your feedback before.

Max

Subject: Re: Cypher package - An extensible Encryption package
Posted by koldo on Tue, 05 Oct 2010 14:28:15 GMT
View Forum Message <> Reply to Message

Hello all

It would be great if Cypher package would include SHA2 functions, like in AESStream, but not depending on OpenSSL.

If you do not know where to get a BSD or better licensed implementation I would do it.

Subject: Re: Cypher package - An extensible Encryption package
Posted by mdelfede on Tue, 05 Oct 2010 20:02:40 GMT
View Forum Message <> Reply to Message

koldo wrote on Tue, 05 October 2010 16:28Hello all

It would be great if Cypher package would include SHA2 functions, like in AESStream, but not depending on OpenSSL.

If you do not know where to get a BSD or better licensed implementation I would do it.

Hi Koldo,

If you find some source about it, I'll be glad to add to cypher package.
I agree with you that we should avoid external dependencies at most.

Ciao

Max

Subject: Re: Cypher package - An extensible Encryption package
Posted by koldo on Wed, 06 Oct 2010 07:00:32 GMT
View Forum Message <> Reply to Message

Hello Massimo and Mindtraveller

A possible source is Crypto++ (http://www.cryptopp.com/
http://en.wikipedia.org/wiki/Crypto%2B%2B).

It is one of the fastest as includes a lot of assembler (for gcc and msvc) and AES-NI, FIPS compliant and using parts of it, it is public domain.

Subject: Re: Cypher package - An extensible Encryption package
Posted by mdelfede on Sat, 09 Oct 2010 22:14:24 GMT
View Forum Message <> Reply to Message

Small change in interface : now base class is Cypher and not CypherBase.
That was needed to make more intuitive the usage of generic Cypher pointers.

Because of that change, virtual function Cypher() (the one that must be redefined in derived classes) now names CypherCypher.

Max

Subject: Re: Cypher package - An extensible Encryption package
Posted by koldo on Fri, 15 Oct 2010 10:12:47 GMT
View Forum Message <> Reply to Message

Hello

Anything about Crypto++?.

May I submit SHA-2 code based on it?

Subject: Re: Cypher package - An extensible Encryption package
Posted by mdelfede on Fri, 15 Oct 2010 10:16:27 GMT
View Forum Message <> Reply to Message

koldo wrote on Fri, 15 October 2010 12:12Hello

Anything about Crypto++?.

May I submit SHA-2 code based on it?

Yes, with pleasure
I've no time to do it right now, if you like to do I'll be happy.
But please, please.... comment it the most you can !

Ciao

Max