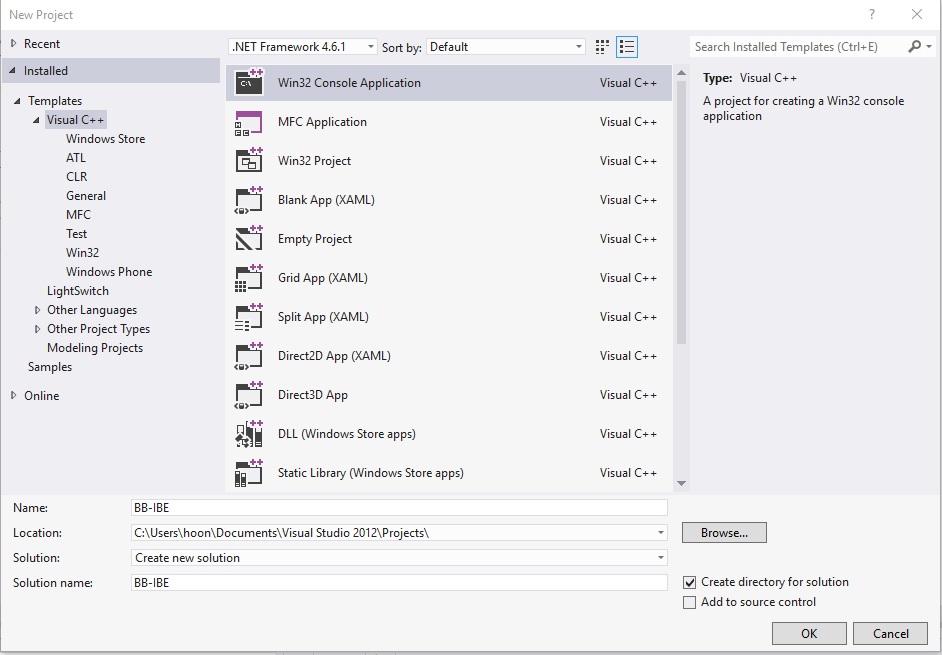
**Report for compiling the BB-IBE (Boneh-Boyen IBE) program:**

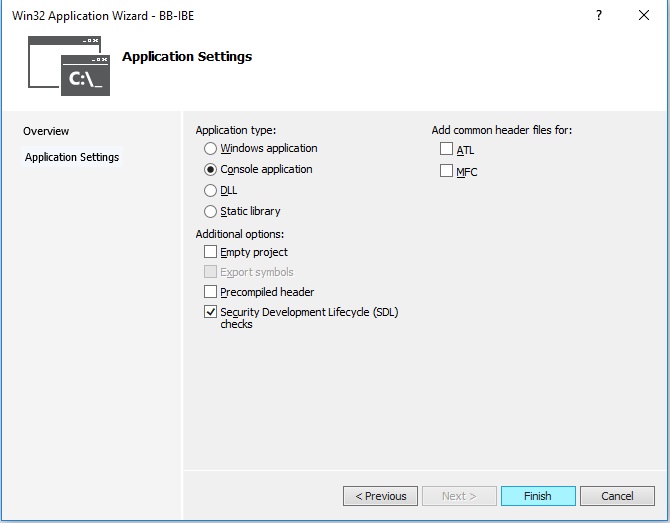
The program was compiled in Visual Studio 2012 and the operating system was Windows 10. In this report we will see how to compile the miracl Boneh-Franklin IBE scheme. Construction of the Boneh-Boyen IBE scheme, called Basic Ident is shown in this report. The scheme shown to be secure in IND-ID-CPA.

* Start with creating a new win 32 Console Application. Write project name “BB-IBE” and solution name “BB-IBE” and click ok.



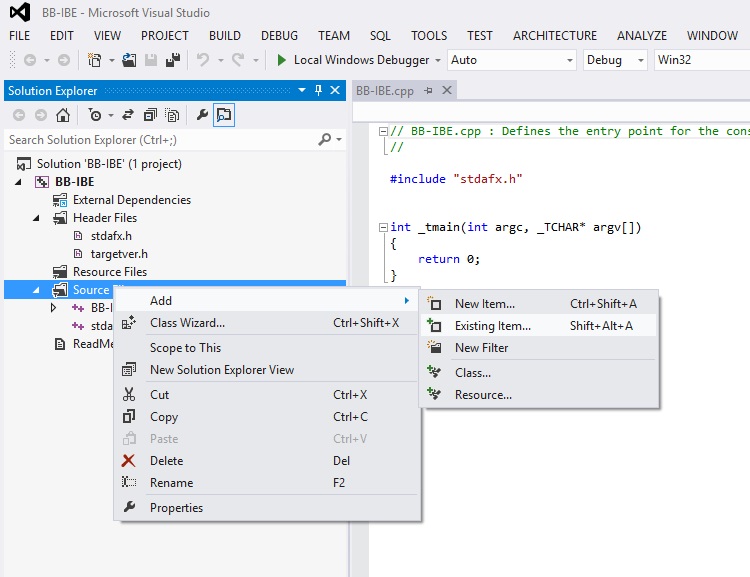
­­

* Now check Console application and Security Development Lifecycle, uncheck the “Precompiler header” in program settings and click finish. You can go to program setting from the left panel as show in the figure below.



* Right Click the project “Source File” in the left panel and got to: Add🡪Existing Item.

Click “Existing Item”.



* Now add the files mentioned below against each type of pairing.
  + For MR\_PAIRING\_SSP curves

ssp\_pair.cpp ecn.cpp zzn2.cpp zzn.cpp big.cpp miracl.lib

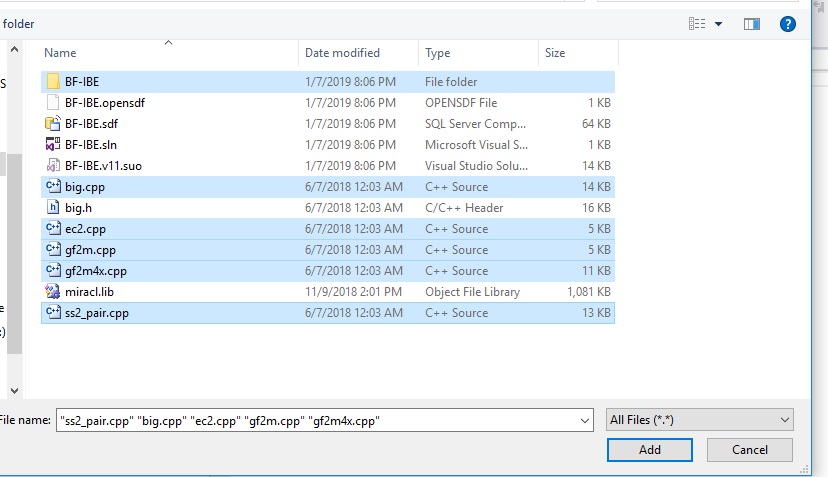
* + For MR\_PAIRING\_SS2 curves

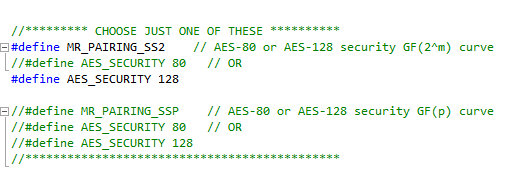
ss2\_pair.cpp ec2.cpp gf2m4x.cpp gf2m.cpp big.cpp miracl.lib

OR

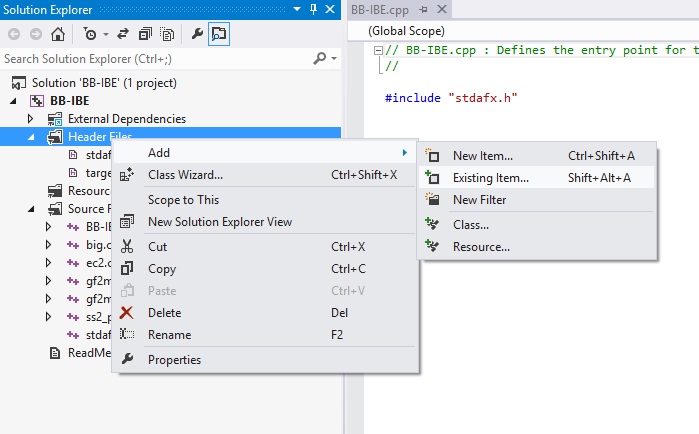
ss2\_pair.cpp ec2.cpp gf2m4x.cpp gf2m.cpp big.cpp miracl.lib

Note: Code for BB-IBE.cpp file is not present in default miracl distribution. Write down all the code provided at the “program code” section of this report, in to the BB-IBE.cpp file.

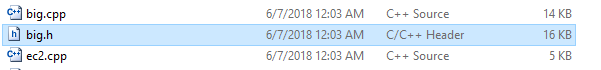


* To choose a pairing do the following:
  + Open the BB-IBE.cpp file. Go to the code section as shown in figure below. (Code has been copied in the BB-IBE.cpp file in above Note).
  + Just uncomment the type of pairing and the security you want to select.
  + Like in this case the pairing that is chosen is “MR\_PAIRING\_SS2” and the security chosen is “AES Security 128”. As shown in the figure.
  + Note that “//” is used to comment in C++, to uncomment just remove the “//” from in front of the line you want to uncomment. Like in this case we have removed “//” at the starting of “#Define MR\_PAIRING\_SS2” and “#Define AES\_SECURITY 128” 
* Right Click the project “Header Files” in the left panel and got to: Add🡪Existing Item. Click

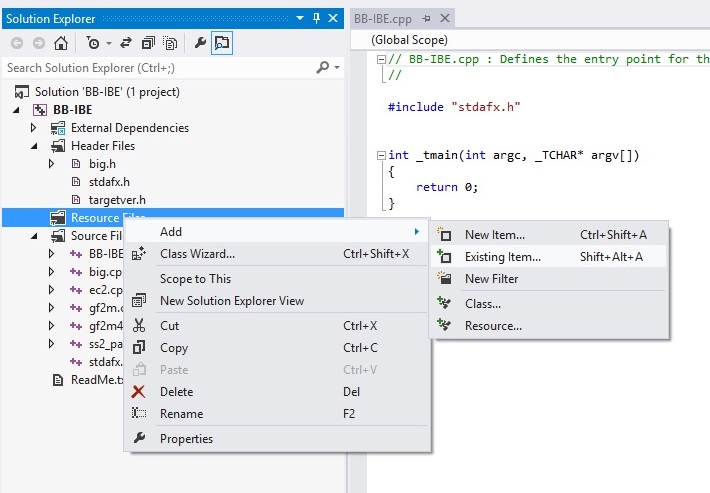
“Existing Item”.

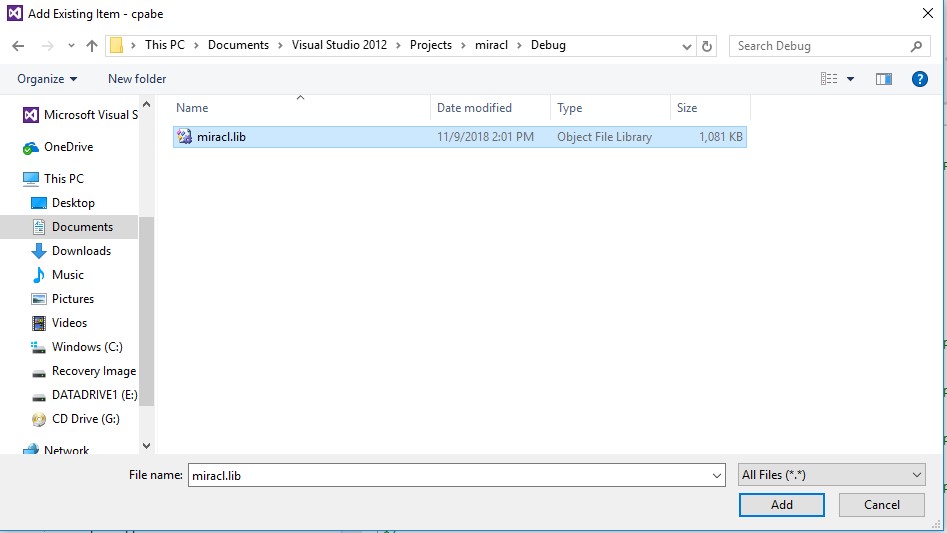


* Include the “big.h” file from miracl distribution.

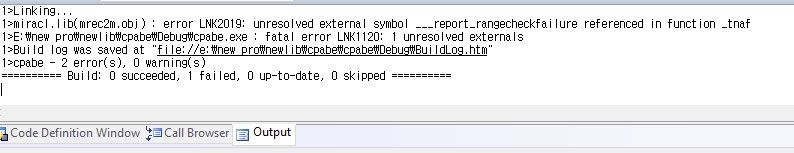


* Add the library file "miracl.lib” in resource folder. Right Click on resource folder and go to ADD🡪Existing Item.

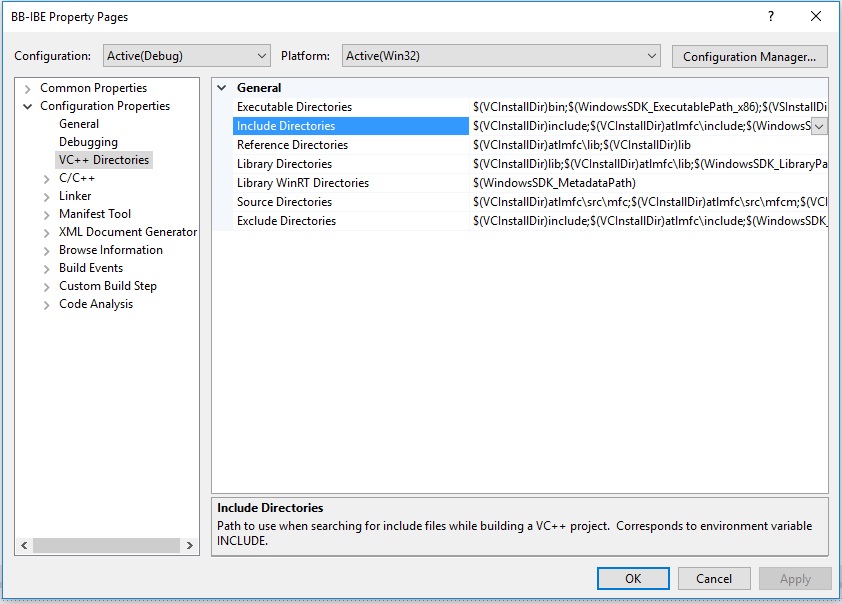
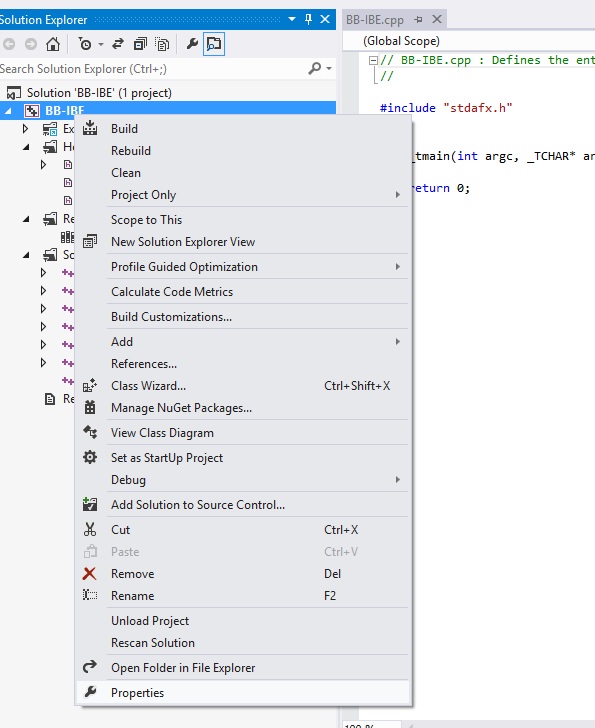




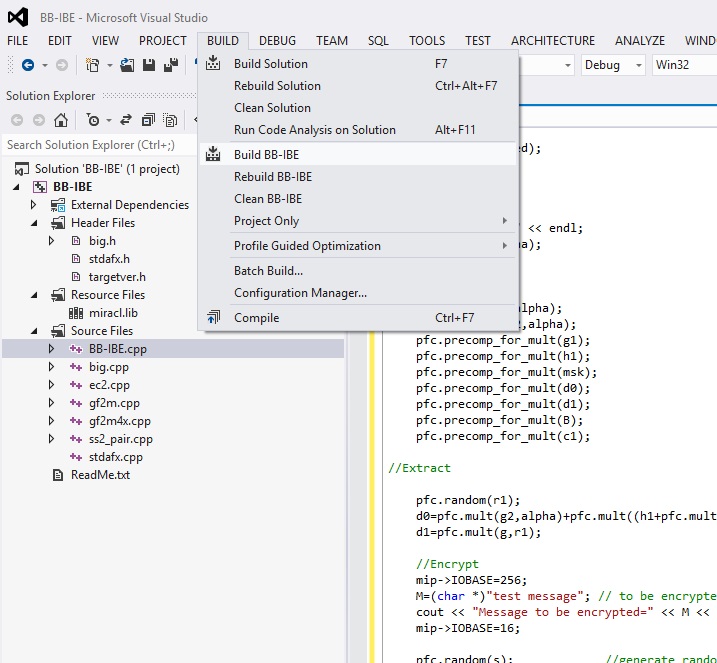
* We added “miracle.lib” to resource files to avoid the errors like error LNK2019, error LNK2001 as shown in figure below.



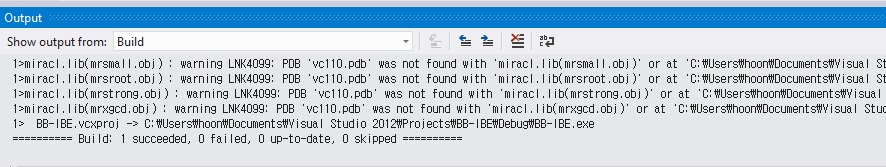
* Go to the properties🡪Configuration settings🡪VC++ directories🡪Include directories. Click on the drop down menu and select the “pairing” folder from miracl distribution.
* Go to the properties🡪Configuration settings🡪VC++ directories🡪Include directories. Click on the drop down menu and select the “include” folder from miracl distribution.
* Go to the properties🡪Configuration settings🡪VC++ directories🡪Include directories. Click on the drop down menu and select the “source” folder from miracl distribution. Then click on APPLY and OK.



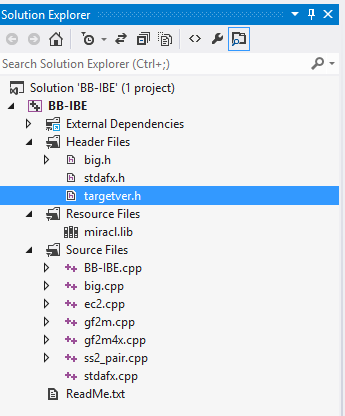
* Build the program from the Build Tab. Click Build BB-IBE.



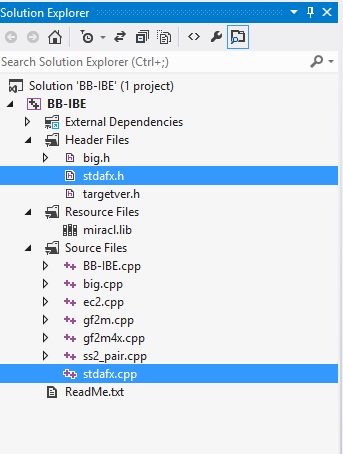
* The program should successfully build like below.



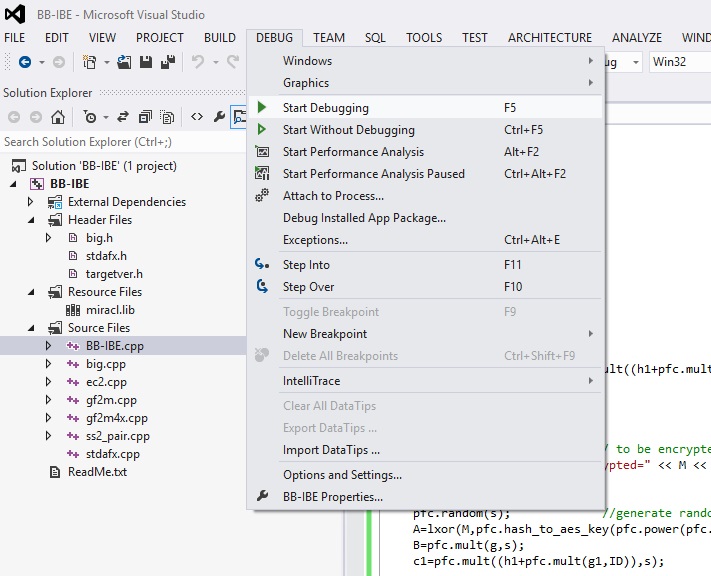
**Note**: From the solution explorer make sure that program has "targetver.h" in header files as show in below figure. As this file Include SDKDDKVer.h defines the highest available Windows platform. If you wish to build your application for a previous Windows platform, include WinSDKVer.h and set the \_WIN32\_WINNT macro to the platform you wish to support before including SDKDDKVer.h. “targetver.h” and “SDKDDKVer.h” are used to control what functions and constants are included into your code from the Windows headers, based on the OS. The “targetver.h” sets defaults to using the latest version of Windows unless thedefines are specified elsewhere. These two files (targetver.h and SDKDDKVer.h ) are auto generated when you create the project, you do not need to manually add them to the project.



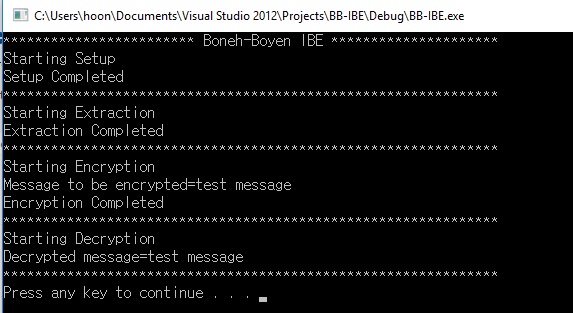
* Also make sure from the solution explorer that program has "stdafx.h" and "stdafx.cpp" as "stdafx.h" as show in figure below. It's a "precompiled header file" any headers you include in stdafx.h are pre-processed to save time during subsequent compilations. These two files (stdafx.cpp and stdafx.h) are auto generated when you create the project, you do not need to manually add them to the project or to create them. (yes it is needed for every console application)



* Now click on Debug and click on Start debugging.

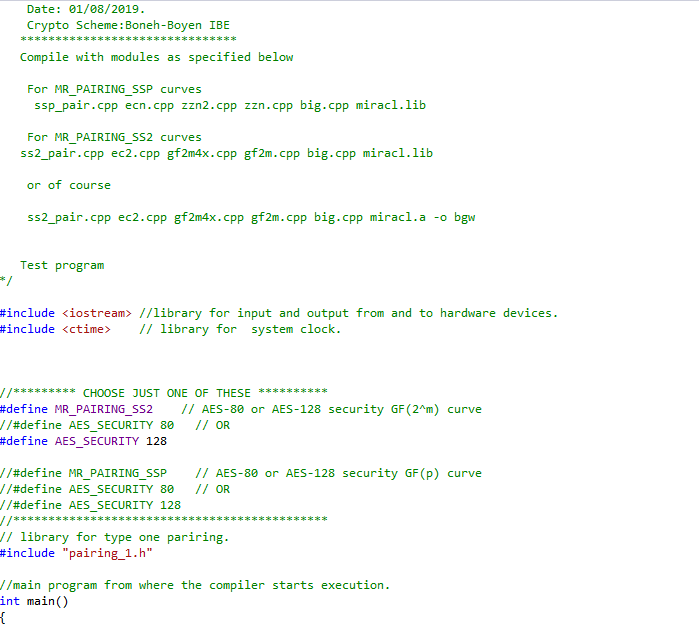
****

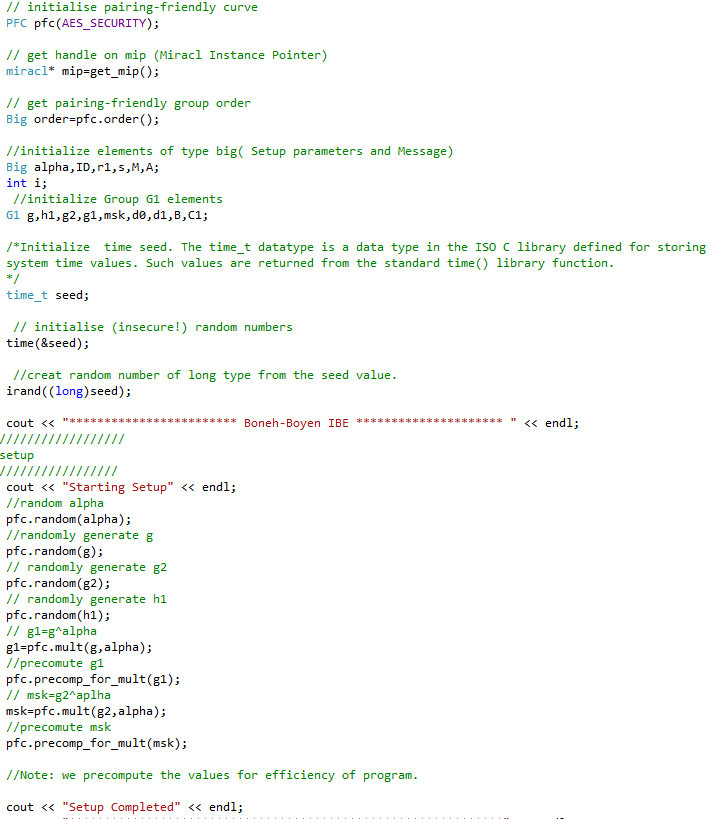
* When you will click on start debugging after successfully building the program. The program will output the following, as shown in figure below.



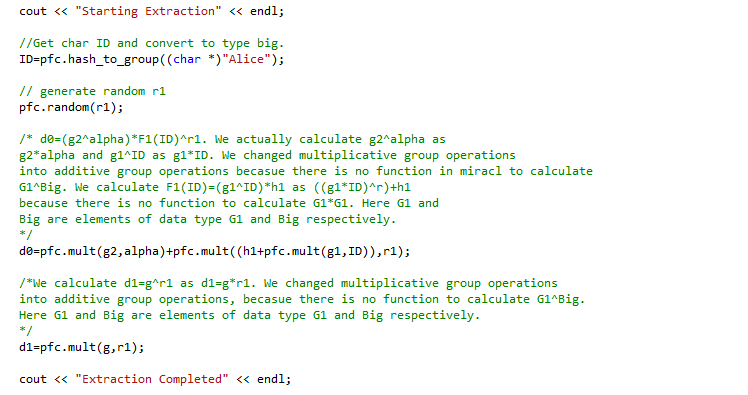
**Program Code**

* Below figure shows the main function of the “BB-IBE” program, where it initializes parameters for the construction of the Boneh-Boyen IBE scheme. The purpose for each parameter (variable) declaration is mentioned against each parameter (variable) in the figure.

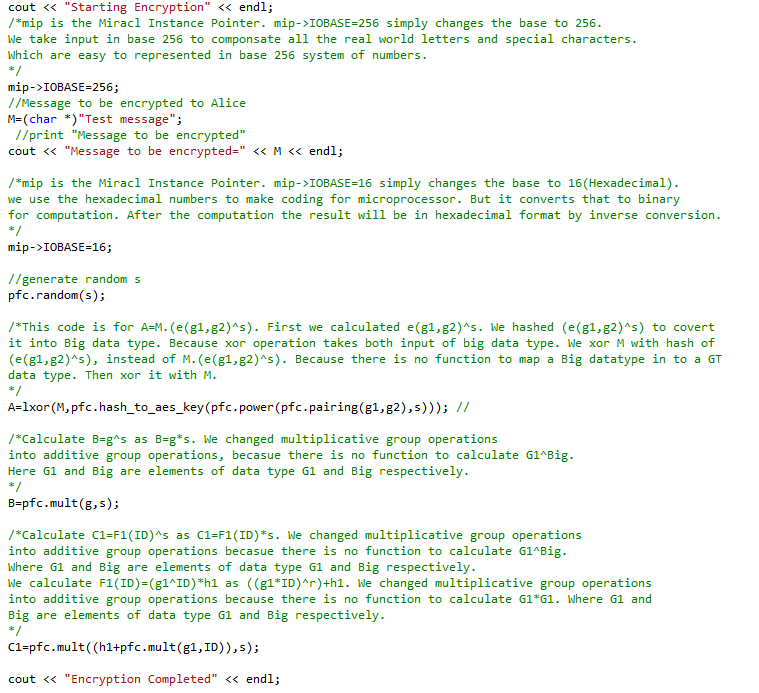




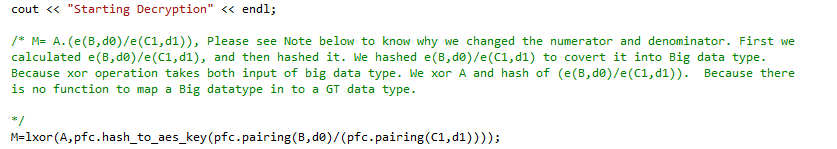
* **Extraction :**



* **Encryption:**



* **Decryption:**



**Note:**

