# PwnHome Research

Pwning your home!



## Guide to HElib (#1)

May 3, 2013 pwnhome Leave a comment Go to comments

Shai Halevi has released an implementation of the BGV cryptosystem over on github. I know a lot of people (myself include) have been wanting to play with FHE to get a better understanding of it. A while back I started a sage implementation of BGV (which I have never released). That helped a lot in understanding BGV. If anyone is interested in the very beta version of my sage code, let me know.

Given that HElib has now been released, I wanted to start playing with it and release my notes. So, here goes. In post #1 I will go over installation and playing with some of the test programs that come with HElib.

### Install (Compilation)

Installation/compilation of HElib is quite easy if you get things right. First of all you will need NTL (number theory library). I tried using the default version of NTL for my version of Ubuntu, but HElib wouldn't compile. So, get the newest version 6.0.0. Compile and install (*./configure; make; make check; make install*).

Then head over to github and clone the git repository (*git clone https://github.com/shaih/HElib.git*). Compilation of HElib is pretty easy (*cd*

*HElib/src; make*). This will build a library against which you can statically compile your own code which can use HElib. There are, however, also some test programs you might be interested in (named *Test_*). To compile these do (*make test*).

## Test_General

Within *Test_General.cpp* we can learn a lot about HElib and how to use it in code. First off is the *main* function. There are a lot of parameters. Running *(./Test_General_x –help*) explains what these parameters are (to some degree). I'll go in more detail on some of them here. The parameters are:

> R is the number of rounds
> p is the plaintext base [default=2]
> r is the lifting [default=1]
> d is the degree of the field extension [default==1]
> (d == 0 => factors[0] defined the extension)
> c is number of columns in the key-switching matrices [default=2]
> k is the security parameter [default=80]
> L is the # of primes in the modulus chain [default=4*R]
> s is the minimum number of slots [default=4]
> m is a specific modulus

*R* is the number of rounds of testing (basically a parameter to a for loop around the testing). *p* is the plaintext base. Basically this is the plaintext space. Setting *p=2* basically says that plaintexts are in {0,1}. *p* should be a prime number. *d* as it says is the degree of the field extension. GHS12 explains (section 2.3) what this is:

> We note that the values in the plaintext slots are not just bits, rather they are polynomials modulo the irreducible F_j's, so they can be used to represents elements in extension fields GF(2^d).

*k* as said, is the security parameter. A default of 80 is 80 bits of security. The length of the modulus chain, *L*, allowes for "leveled" FHE. *L* reduces or eliminates the amount of bootstrapping (see BGV11). Finally, *s* tells how many (at a minimum) number of plaintext slots you'd like. Set this too high and it will complain.

The basic parameters given in the file are pretty good for testing. Take a look at the GHS12 paper for more on parameter tuning. In that paper they look specifically at

tuning BGV for AES.

★ Like

Be the first to like this.

Categories: Cryptography, HElib Tags: homomorphic encryption
Comments (21) Trackbacks (0) Leave a comment Trackback

1. 
   David
   May 7, 2013 at 9:23 am | #1
   Reply | Quote

   I am trying to compile HElib on Windows using VS8, but so far no success. Can you add a section detailing how to get the library working on Windows? Thx.

   - 
     pwnhome
     May 7, 2013 at 5:19 pm | #2
     Reply | Quote

     Probably won't post a comprehensive section on HElib for Windows any time soon (hardly use Windows these days). Do you have NTL built? They have Windows instructions. What errors are you getting?

     - 
       David
       May 7, 2013 at 8:16 pm | #3
       Quote

       Yes, I managet to get NTL compiled on windows. However, HElib is not compiling. Here are some of the error I am getting:
       error C2668: 'log' : ambiguous call to overloaded function
       error C2065: '__func__' : undeclared identifie
       error C2065: '__func__' : undeclared identifier
       error C2065: 'M_PI' : undeclared identifier
       error C2039: 'zMstar' : is not a member of 'FHEcontext'

error C2039: ‘M’ : is not a member of ‘PAlgebra’
error C2065: ‘M_PI’ : undeclared identifier
error C2039: ‘zMstar’ : is not a member of ‘FHEcontext’

When I created the HElib project in VS8, I simply imported the .cpp and .h files and referenced the ntl.lib library.
Is there extra configurations in VS8 that need to be added?

- 

  pwnhome
  May 8, 2013 at 5:52 am | #4
  Quote

  Did you add the NTL header files to VS8? As HElib uses NTL, the compiler needs to know where the NTL header files are. Are there file names and line numbers associated with those error messages? For example, where is the call to ‘log’ that is referenced in the first error?

2. 

   David
   May 10, 2013 at 5:45 pm | #5
   Reply | Quote

   Yes, I already added the NTL header file to the project.
   Here are more details about the errors:
   Error 6 error C2668: ‘log’ : ambiguous call to overloaded function c:\documents and settings\daghir\my documents\visual studio 2008\projects\helib\helib-master\src\cgauss.cpp 89 HElib

   Error 24 error C2065: ‘__func__’ : undeclared identifier c:\documents and settings\daghir\my documents\visual studio 2008\projects\helib\helib-master\src\cmodulus.cpp 152 HElib

   Error 4 error C2065: ‘M_PI’ : undeclared identifier c:\documents and settings\daghir\my documents\visual studio 2008\projects\helib\helib-master\src\cgauss.cpp 33 HElib

Error 109 error C2039: ‘zMstar’ : is not a member of ‘FHEcontext’ c:\documents and settings\daghir\my documents\visual studio 2008\projects\helib\helib-master\src\old-test_fhe.cpp 18 HElib

Error 110 error C2039: ‘M’ : is not a member of ‘PAlgebra’ c:\documents and settings\daghir\my documents\visual studio 2008\projects\helib\helib-master\src\old-test_fhe.cpp 34 HElib

- 

  pwnhome
  May 27, 2013 at 12:32 pm | #6
  Reply | Quote

  Not really sure what is up and I haven’t had any time to play with compiling the library in Windows. Good luck though. Sorry I couldn’t help more.

3. 

   George
   May 28, 2013 at 6:33 am | #7
   Reply | Quote

   Dear pwnhoem,

   I was also trying to compile HElib in Linux, and what get is
   —
   In file included from bluestein.cpp:33:
   bluestein.h:64: error: expected initializer before ‘<’ token
   compilation terminated due to -Wfatal-errors.
   make: *** [bluestein.o] Error 1
   —
   Do you have any ideas what could cause it?

   Cheers.

   - 

     pwnhome
     May 28, 2013 at 6:42 am | #8

Reply | Quote

I believe I had the same problem and it had to do with not having the proper version of NTL installed. Make sure you have NTL 6.0.0 (see link in my post).

4. 

George
May 28, 2013 at 7:20 am | #9
Reply | Quote

Dear pwnhome,

thank you for your quick reply! I installed NTL 6.0.0, just as you advised in your post. And when I run “make check”, I see something like … for all the tests.
—
making QuadTest
make[1]: Entering directory `/home/George/Downloads/ntl-6.0.0/src'
g++ -I../include -I. -O2 -o QuadTest QuadTest.c ntl.a -lm #LSTAT
make[1]: Leaving directory `/home/George/Downloads/ntl-6.0.0/src'
running QuadTest
QuadTest OK
—

But I still get the following error every time I try to install HElib:
—
~/Downloads/HElib-master/src$ make
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c bluestein.cpp
In file included from bluestein.cpp:33:
bluestein.h:64: error: expected initializer before ‘<’ token
compilation terminated due to -Wfatal-errors.
make: *** [bluestein.o] Error 1
—

5. 

George
May 28, 2013 at 8:46 am | #10

Reply | Quote

upd: was an internal error. Solved it by adding to the CFLAGS (line 19 in the Makefile) the exact path to where NTL is installed in:
CFLAGS = -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -I$$HOME/Downloads/ntl-6.0.0/include/

That solved it! Had this problem only because I couldn't (didn't have root rights) install NTL in the standard /usr/local directory.

- 

  pwnhome
  May 28, 2013 at 9:07 am | #11
  Reply | Quote

  Glad you got it working. I was going to guess that there was something wrong with how NTL header files were being included.

6. 

   George
   May 29, 2013 at 3:42 am | #12
   Reply | Quote

   Excited to see further articles in the "Guide to HElib" series!

7. 

   ram
   June 19, 2013 at 4:11 am | #13
   Reply | Quote

   Dear pwnhome,
   I installed NTL 6.0.0 successfully.But Iam getting following errors when I am trying to "make" HElib/src…..Requesting help in this issue.
   Thank you,

   g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c NumbTh.cpp
   g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c timing.cpp

```
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c bluestein.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c PAlgebra.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c
CModulus.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c
FHEContext.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c IndexSet.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c
DoubleCRT.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c
SingleCRT.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c FHE.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c
KeySwitching.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c Ctxt.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c
EncryptedArray.cpp
g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -c replicate.cpp
ar ruv fhe.a NumbTh.o timing.o bluestein.o PAlgebra.o CModulus.o
FHEContext.o IndexSet.o DoubleCRT.o SingleCRT.o FHE.o KeySwitching.o Ctxt.o
EncryptedArray.o replicate.o
ar: creating fhe.a
a – NumbTh.o
a – timing.o
a – bluestein.o
a – PAlgebra.o
a – CModulus.o
a – FHEContext.o
a – IndexSet.o
a – DoubleCRT.o
a – SingleCRT.o
a – FHE.o
a – KeySwitching.o
a – Ctxt.o
a – EncryptedArray.o
a – replicate.o
```

- 

  pwnhome
  June 19, 2013 at 5:41 am | #14
  Reply | Quote

  Those don't look like errors at all. It looks like everything has compiled properly. Do you have an fhe.a file in the directory after this is run? If so, everything is correct.

8. 

   ram
   June 19, 2013 at 6:33 am | #15
   Reply | Quote

   Thanks for quick reply.I have fhe.h file but when I execute " make test" it gave me like this:

   g++ -g -O2 -Wfatal-errors -Wshadow -Wall -I/usr/local/include -o Test_General_x Test_General.cpp fhe.a -L/usr/local/lib -lntl -lgmp -lm
   /usr/bin/ld: cannot find -lgmp
   collect2: error: ld returned 1 exit status
   make: *** [Test_General_x] Error 1

   due to which I thought there was ERROR in "make".
   Thank you
   cheers

   - 

     pwnhome
     June 19, 2013 at 6:47 am | #16
     Reply | Quote

     Okay, that is a problem. You need libgmp. Should be available in your package repositories. Might need the -dev version too. In ubuntu, the two packages are libgmp10 and libgmp-dev.

9. 

ram
June 19, 2013 at 6:50 am | #17
Reply | Quote

okay …I will do that…Thank you very much. waiting to see more articles from you.
Thank you

10. 
ram
June 24, 2013 at 6:09 am | #18
Reply | Quote

Dear pwnhome,
Do I need to edit my " Makefile " when I compile my programs using HElib…I am not good at it…so Can you please help me know how to edit my MakeFile to link my program while compiling..
Thanks in advance,

- 
pwnhome
June 24, 2013 at 2:29 pm | #19
Reply | Quote

Which Makefile? Did you create your own or are you using the one from post #3? If using the one from post #3, yes you'll have to modify it and the details are in the post.

11. 
ram
June 24, 2013 at 10:15 pm | #20
Reply | Quote

thanks .I didn't see your post#3….That's what I actually need.sorry for troubling you .

12. 

nixfreak
July 15, 2013 at 1:52 pm | #21
Reply | Quote

Compile HElib on Ubuntu 12
1. Install build-essential, libgmp3c2 and libgmp3-dev
2. download latested NTL – cd into /src and ./configure, make ,make check, make install
3. Compile HElib download zip or clone the master; cd /src and make, make test.

Should have a working compiled version of HElib now.

1. No trackbacks yet.

## Leave a Reply

Enter your comment here...

Analysis of XKCD's Hash Competition CipherCloud DMCA
RSS feed

## Projects

- CourseMaps
- Intro to GNURadio and USRP Code
- The Homomorphic Encryption Project

## Tag Cloud

AES cryptanalysis Cryptography exploitation FFT filesystem firmware GNURadio hacking homomorphic encryption iPhone jailbreaking MITM objective-c paillier Penetration Testing Proxy Quantum reversing rootkits shellcode Source source code ssh SSL thep USRP Windows Mobile

## Categories

- Cryptography
- General
- GNURadio and USRP
- HElib
- iPhone
- thep
- Uncategorized

## Archives

- June 2013
- May 2013
- April 2013
- February 2012
- November 2011
- August 2011
- April 2011
- January 2011
- November 2010
- May 2010
- April 2010
- March 2010
- October 2009
- September 2009
- August 2009
- July 2009

- June 2009
- May 2009
- April 2009
- March 2009

Top WordPress
Blog at WordPress.com. The INove Theme.