INSTALLATION GUIDE FOR MOTION2NX

Table of Contents

[**WHAT IS MOTION2NX?**](#_heading=h.30j0zll) **1**

[**SYSTEM REQUIREMENTS**](#_heading=h.1fob9te) **1**

[**INSTALLING THE REPOSITORY**](#_heading=h.3znysh7) **7**

# WHAT IS MOTION2NX?

* Massive Repository used to implement Secure Multi-party Computation using the ABY2.0 protocol. ([paper](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiusOWHn_b3AhWcTGwGHWdLBNYQFnoECAgQAQ&url=https%3A%2F%2Feprint.iacr.org%2F2020%2F1225.pdf&usg=AOvVaw1jCz_7QKr70D60tzMWQe_x))
* Consists of multiple in-built examples and benchmarking files
* Implementation of Yao’s millionaire’s problem can be used to check whether the installation is functioning or not.
* The repository can be found [HERE](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwis9Oq2ofb3AhWxjuYKHSJwD4wQFnoECAsQAQ&url=https%3A%2F%2Fgithub.com%2Fencryptogroup%2FMOTION2NX&usg=AOvVaw1TSQgtsdfPqj2L4v_Y4ZlK).

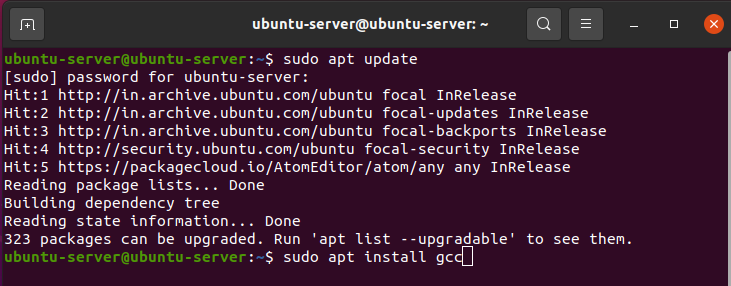
# SYSTEM REQUIREMENTS

1. **A basic PC setup running on any linux operating system** 
   * If you do not have it, you can either get a dual-booted system with your current OS or you can download a virtual machine such as [VirtualBox](https://www.virtualbox.org/) or [VMWare](https://www.vmware.com/pdf/view45_installation_guide.pdf).
   * For both cases, you will need a Linux OS. The most commonly used distribution would be Ubuntu. The various versions for downloading can be found [HERE](https://ubuntu.com/desktop).
   * Follow the steps in each of these downloading documentations and you should have a linux system up and running!
2. **C++ downloaded on the linux system**
   * Though all linux systems come with an in-built gcc and g++ compiler, it is always useful to check if it is present
   * Open the terminal (On the sidebar or use ctrl+T) and type in g++ –version . The latest version should ideally be above 9 for gg or g++.
   * If you wish to install or reinstall it, simply perform:

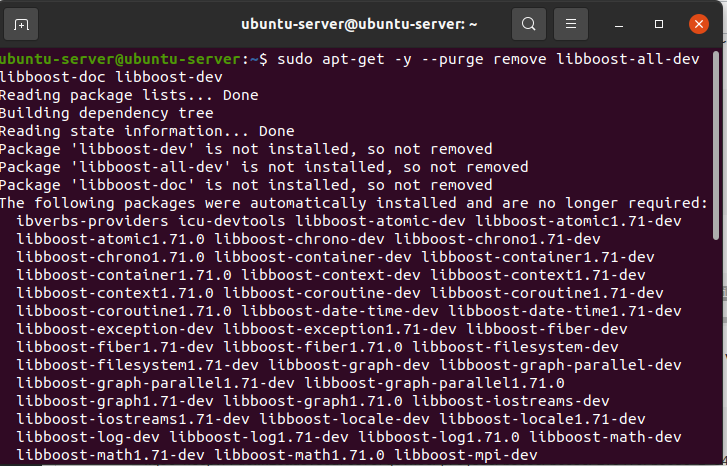
sudo apt update

sudo apt install g++

sudo apt update



1. **Downloading and updating boost libraries**
   * This is a slightly tricky part since most linux systems do not come with the latest boost updates
   * You can first check if the current boost version is accepted by building the repository in section III. If it is not supported, then the following steps must be done.
   * The first step would be to uninstall the existing boost libraries present on the system. This can be done using:



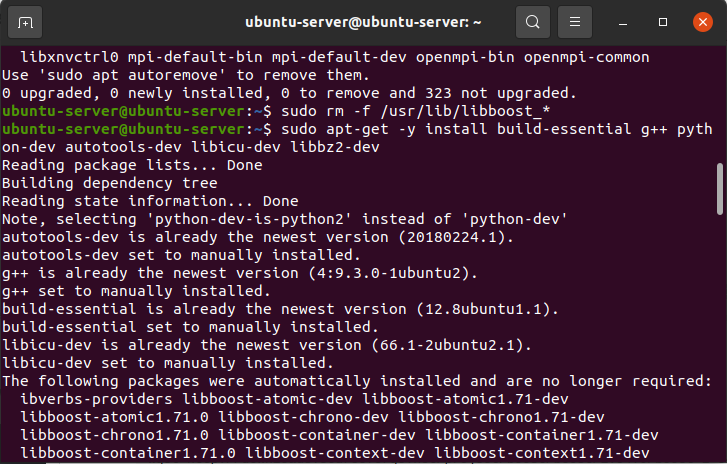
sudo apt-get update

sudo apt-get -y --purge remove libboost-all-dev libboost-doc libboost-dev

sudo rm -f /usr/lib/libboost\_\*

* + The above code only deletes boost and not its dependencies. We shall need the dependencies to install the new version.
  + For the latest versions of boost, we might need to install additional dependencies, which can be done using:

sudo apt-get -y install build-essential g++ python-dev autotools-dev libicu-dev libbz2-dev



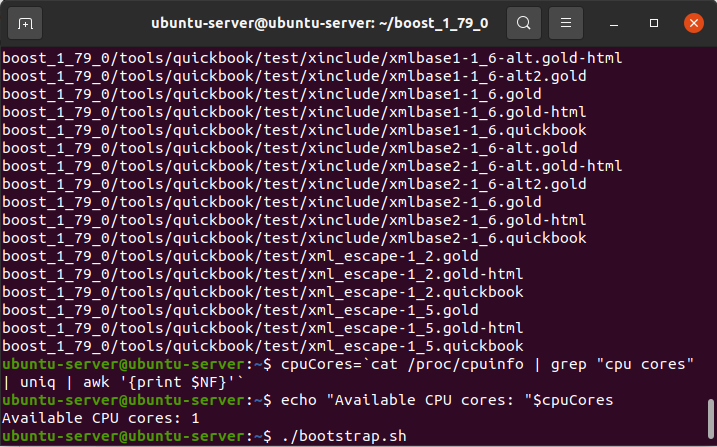
* + Once we have this setup, go to the [boost downloads](https://www.boost.org/users/history/version_1_79_0.html) and download the latest version of boost. Make sure to install the .tar.gz extension as it is easier to install with. In this case, we are downloading 1.79 [HERE](https://boostorg.jfrog.io/artifactory/main/release/1.79.0/source/boost_1_79_0.tar.gz).
  + Go to the place where the download is present and simply execute this to unzip the folder, then cd into the folder:

tar -zxvf boost\_1\_79\_0.tar.gz

* + We can check the number of CPU cores free. (Apparently makes the installation faster):

cpuCores=`cat /proc/cpuinfo | grep "cpu cores" | uniq | awk '{print $NF}'`

echo "Available CPU cores: "$cpuCores

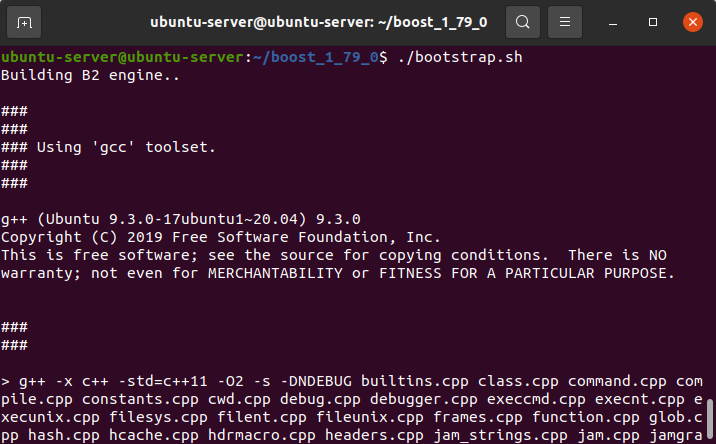


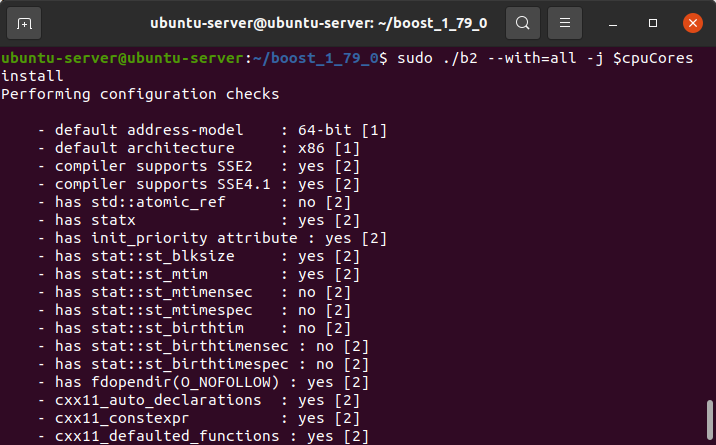
* + Now, execute these one after the other:

./bootstrap.sh --with-libraries=atomic,date\_time,exception,filesystem,iostreams,locale,program\_options,regex,signals,system,test,thread,timer,log

# this will generate ./b2

sudo ./b2 --with=all -j $cpuCores install





* + This might take around an hour to get the entire boost setup on the PC.
  + To check if it is installed successfully, we simply type in:

cat /usr/local/include/boost/version.hpp | grep "BOOST\_LIB\_VERSION"

* + Now you have successfully installed the latest version of boost on your PC!

1. **Installing git**
   * Easiest of all tasks, if you do not have git simply sudo update and then execute install git as follows:

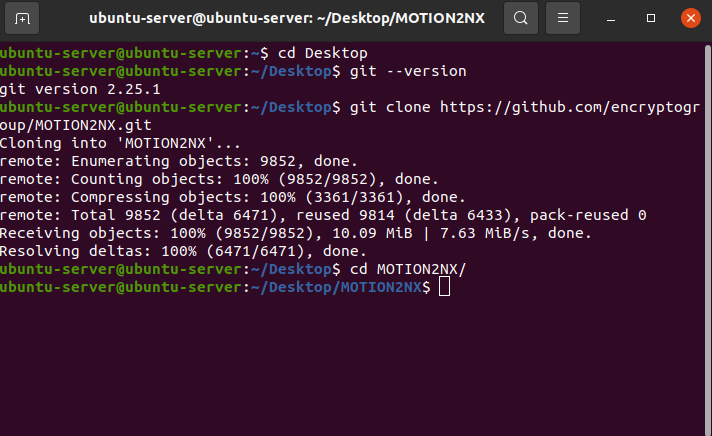
sudo apt update

sudo apt install git

# INSTALLING THE REPOSITORY

* On your linux system, move into the folder you wish to install your MOTION2NX repository on.
* Go to the [github repository](https://github.com/encryptogroup/MOTION2NX) and click on the green code dropdown. This gives you the cloning options.
* Select the HTTPS option and copy the link. [HERE](https://github.com/encryptogroup/MOTION2NX.git) is the shortcut.
* Go to the terminal and cd to the folder of choice.
* Execute a clone command as follows:

git clone https://github.com/encryptogroup/MOTION2NX.git



* Now cd into the repository and copy paste the entire code below at once execute:

CC=gcc CXX=g++ cmake \

-B build\_debwithrelinfo\_gcc \

-DCMAKE\_BUILD\_TYPE=DebWithRelInfo \

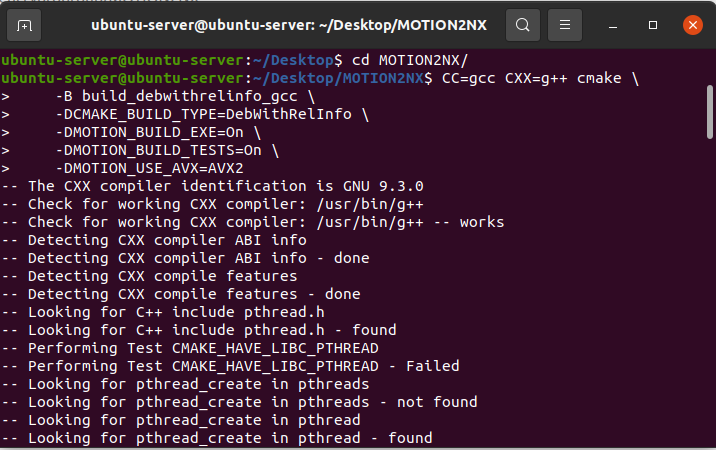
-DMOTION\_BUILD\_EXE=On \

-DMOTION\_BUILD\_TESTS=On \

-DMOTION\_USE\_AVX=AVX2

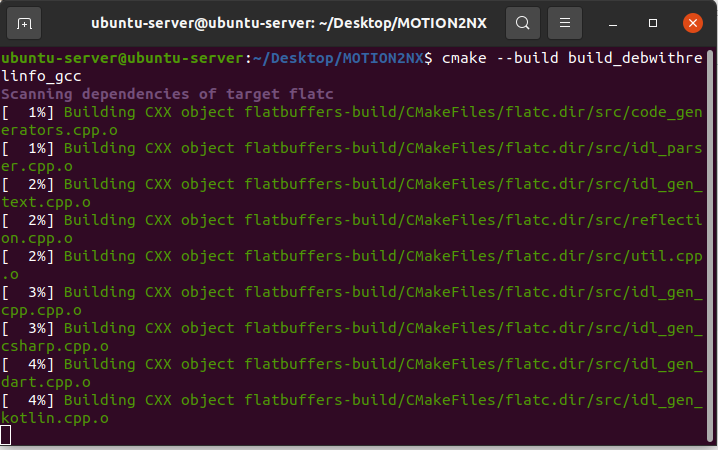
my : ghp\_YAInRwiLuZx2kDKg0xtOuAaHUM27qz456IQC

ghp\_twUmiq0qVbenaQqQZS3TpJwjPKf3IF2V7mqc



* The above code builds the necessary folder, and each of the flags indicate the requirements to be fulfilled for the build.
* Once that is done, execute the command to install the executables and their dependencies. This process takes another hour worst case:

cmake --build build\_debwithrelinfo\_gcc



* Now, cd into build\_deb…. Repository and execute the following commands to test the millionaire’s problem:
  + Party1:

./bin/millionaires\_problem --my-id 0 --party 0,::1,7000 --party 1,::1,7001 --arithmetic-protocol beavy --boolean-protocol yao --repetitions 5 --input-value 42

* + Party2 :

./bin/millionaires\_problem --my-id 1 --party 0,::1,7000 --party 1,::1,7001 --arithmetic-protocol beavy --boolean-protocol yao --repetitions 5 --input-value 47 --json

* Congratulations! You now have a working ABY2.0 SMPC model that runs the Yao’s millionaire’s problem!