**Installation Guide: BIKE**

This README outlines the necessary steps to set up and run Part 2 of the cryptography project, which involves the BIKE key encapsulation mechanism. Follow these instructions to unzip the required files, establish a Python virtual environment, install necessary dependencies, and execute the application components.

**Prerequisites**

- Python 3.6 or higher

- pip (Python package installer, usually included with Python 3.4 and above)

- Access to a terminal or command line interface

**Setup Instructions**

**1. Extract the BIKE Folder**

Ensure you have the BIKE.zip file in your working directory. Use the following command to unzip the folder:

unzip BIKE.zip

cd BIKE

**2. Create Python Virtual Environment**

A virtual environment is recommended to manage dependencies without affecting global Python settings.

#Install virtualenv if not already installed

pip install virtualenv

# Create a virtual environment

virtualenv venv

# Activate the virtual environment

source venv/bin/activate # On Windows use .\venv\Scripts\activate

**3. Install Required Packages**

Install the required libraries, including liboqs and the cryptography package.

# Ensure pip, setuptools, and wheel are up to date

pip install --upgrade pip setuptools wheel

**Installation**

Configure, build and install liboqs

Execute in a Terminal/Console/Administrator Command Prompt

git clone --depth=1 https://github.com/open-quantum-safe/liboqs

cmake -S liboqs -B liboqs/build -DBUILD\_SHARED\_LIBS=ON

cmake --build liboqs/build --parallel 8

cmake --build liboqs/build --target install

The last line may require prefixing it by sudo on UNIX-like systems. Change --parallel 8 to match the number of available cores on your system.

On UNIX-like platforms, you may need to set the LD\_LIBRARY\_PATH (DYLD\_LIBRARY\_PATH on macOS) environment variable to point to the path to liboqs' library directory, e.g.,

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/lib

On Windows platforms, you must ensure that you add the -DCMAKE\_WINDOWS\_EXPORT\_ALL\_SYMBOLS=TRUE flag to CMake, and that the liboqs shared library oqs.dll is visible system-wide, i.e., set the PATH environment variable accordingly by using the "Edit the system environment variables" Control Panel tool or executing in a Command Prompt

set PATH=%PATH%;C:\Program Files (x86)\liboqs\bin

You can change liboqs' installation directory by configuring the build to use an alternative path, e.g., C:\liboqs, by passing the -DCMAKE\_INSTALL\_PREFIX=/path/to/liboqs flag to CMake, e.g.,

cmake -S liboqs -B liboqs/build -DCMAKE\_INSTALL\_PREFIX="C:\liboqs" -DCMAKE\_WINDOWS\_EXPORT\_ALL\_SYMBOLS=TRUE -DBUILD\_SHARED\_LIBS=ON

Let liboqs-python install liboqs automatically

If liboqs is not detected at runtime by liboqs-python, it will be downloaded, configured and installed automatically (as a shared library). This process will be performed only once, at runtime, i.e., when loading the liboqs-python wrapper. The liboqs source directory will be automatically removed at the end of the process.

This is convenient in case you want to avoid installing liboqs manually, as described in the subsection above.

**Install and activate a Python virtual environment**

Execute in a Terminal/Console/Administrator Command Prompt

python3 -m venv venv

. venv/bin/activate

python3 -m ensurepip --upgrade

On Windows, replace the line

. venv/bin/activate

by

venv\Scripts\activate.bat

**Configure and install the wrapper**

Execute in a Terminal/Console/Administrator Command Prompt

git clone --depth=1 https://github.com/open-quantum-safe/liboqs-python

cd liboqs-python

pip install .

Run the examples

Execute

python3 liboqs-python/examples/kem.py

python3 liboqs-python/examples/sig.py

python3 liboqs-python/examples/rand.py

Run the unit test

Execute

nose2 --verbose liboqs-python

# Install cryptography package

pip install cryptography

**Running the Application**

**Start the Server**

To initiate the server, open a terminal and execute the following command:

python server.py

**Start the Client**

In a new terminal, start the client by running:

python client.py

**Running Efficiency Tests**

BIKE Tests

Run the BIKE efficiency tests with the following command:

python Testing\_BIKE.py

**Additional Notes**

* Ensure that all commands are compatible with your operating system, particularly when activating the virtual environment or installing packages requiring compilation.
* Update your Python and pip regularly to maintain compatibility with new library versions.

**Troubleshooting**

* If you encounter issues with pip commands, verify that the virtual environment is active and that pip is updated.
* If there are errors during the installation of liboqs-python, check for the necessary C compiler and Python development headers in your system.