Quantum key distribution fast post-processing Focus on fast Privacy Amplification on GPUs

Nico Bosshard, Forschungsmitarbeit FS 2021
Betreuung: Dr. Esther Hänggi, Roland Christen
Hochschule Luzern, Informatik, 11. Juni 2021

Quick-Guide

Requirements:

Minimum hardware requirements:
Multicore x64 CPU (Testet on AMD Ryzen Threadripper 3970X 32 Core/64 Thread CPU)
16 GB RAM (Testet on 128 GB DDR4 ECC M391A2K43BB1-CTD)
8 GB GPU Memory
NVidia GPU (Tested on RTX 3080, RTX 2070 Super)
AMD GPU (Tested on RX 5700 XT)

Preparations Windows:
Visual Studio 2019: https://visualstudio.microsoft.com/de/vs/
CMake: https://cmake.org/download/
NVIDIA GPU Computing Toolkit v11.3: https://developer.nvidia.com/cuda-downloads
Vulkan SDK 1.2.170.0: https://vulkan.lunarg.com/
Git for Windows: https://git-scm.com/download/win
or GitHub Desktop: https://desktop.github.com/
Python 3: https://www.python.org/downloads/

Preparations Linux:
nvidia-docker: https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html
git (sudo apt-get install git)
python (sudo apt-get install python3)

Edit config:
Open config.yaml using a text editor. Carefully read the description inside the config file.

Azure Pipeline:

Azure Pipeline:
See azure-pipelines.yml
Current pipeline is on https://nicoboss.visualstudio.com/PrivacyAmplification
The pipeline gets triggered when pushing to: https://github.com/nicoboss/PrivacyAmplification and

NOT https://gitlab.enterpriselab.ch/qkd/gpu/privacyamplification - make sure to push to booth of them at the same time. To do so configure git like this by editing .get/config:

```
[remote "origin"]
	url = git@gitlab.enterpriselab.ch:qkd/gpu/privacyamplification.git
	url = git@github.com:nicoboss/PrivacyAmplification.git
	fetch = +refs/heads/*:refs/remotes/origin/*
```

I recommend using PGP and SSH keys. Ask me for access to GitHub and Azure DevOps.

Create Deliverables:
The Pipeline automatically generates portable deliverables using pipeline artefacts. Those can be downloaded from Azure DevOps. After the download run downloadAssets.cmd to get the required Assers. The NSIS installer is depreciated and should no longer be used.

Upload Assets (Azure Artefacts):

Edit privacyamplification\PrivacyAmplification\Scripts\uploadAssets.cmd and make sure to also edit privacyamplification\PrivacyAmplification\Scripts\downloadAssets.cmd

to modify the new version number. Create a subfolder containing only the single Artefact file to upload for every artefact and run uploadAssets.cmd.

Git:

Clone git:
Private: git clone --recursive git@gitlab.enterpriselab.ch:qkd/gpu/privacyamplification.git
Public: git clone --recursive https://oauth2:3_wtn2uTreSXt9q1ihXV@gitlab.enterpriselab.ch/qkd/gpu/privacyamplification.git

Main Branch:
master: Just use this one.

Other interesting branches:
subblocks: Contains the subblock code for small blocks. Needs to be fixed, tested and properly merged. Redoing the whole thing in this branch with separate GPU Code so it doesn't impact the main algorithm is recommended.
half_accuracy_vulkan: Proof of concept half Precision Vulkan algorithm. Should be revisited once VkFFT fixes the issue where it can't perform a mixed precision FFT.

Libraries:

Used Libraries:

https://github.com/DTolm/VkFFT

https://github.com/jgbit/vuda

https://github.com/zeromq/libzmq

https://github.com/KhronosGroup/glslang

https://github.com/rhash/RHash

https://github.com/progschj/ThreadPool

https://github.com/jimmiebergmann/mini-yaml

http://half.sourceforge.net/ (currently unused)

Update Vulkan:

Just download and install the latest Vulkan SDK from https://vulkan.lunarg.com/

Update Cuda:

Uninstall every Nvidia related Application and driver
Install latest NVIDIA GPU Computing Toolkit from https://developer.nvidia.com/cuda-downloads
Change the NVIDIA GPU Computing Toolkit version number inside every *.vcxproj File

The new cuFFT DLL located at C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\vXX\bin\cufft64_10.dll needs to be uploaded as Azure Artifact. Follow **Upload Assets** for this.

How to update VkFFT:

Replace current vkFFT.h with https://github.com/DTolm/VkFFT/blob/master/vkFFT/vkFFT.h
I recommend doing this every few days as VkFFT is in very active development.

Update libzmq:

Every few months it's recommended to update libzmq. To do so perform the following stps:

git clone --recursive https://github.com/zeromq/libzmq.git
cd libzmq
Delete everything inside privacyamplification\libzmq\include
Copy everything inside include to privacyamplification\libzmq\include
mkdir build
cd build
cmake ..
ZeroMQ.sln

Build Release|x64
cd bin/Release
Delete everything inside privacyamplification\libzmq\dll\
Copy libzmq-v142-mt-4_X_X.dll to privacyamplification\libzmq\dll\
cd ..\..\lib\Release
Delete everything inside privacyamplification\libzmq\lib\
Copy libzmq-v142-mt-4_X_X.exp to privacyamplification\libzmq\lib\
Copy libzmq-v142-mt-4_X_X.lib to privacyamplification\libzmq\lib\
libzmq-v142-mt-s-4_X_X.lib to privacyamplification\libzmq\lib\
Change the libzmq version number inside every *.vcxproj File

The created DLL also needs to be uploaded as Azure Artifact. Follow **Upload Assets** for this.

Update vuda:
Because I modified Vuda quite a lot changes from the official repository needs to be compared and manually merged.

Update glslang:
I recommend to just use the version inside the VkFFT repository but the official version should work too. Glslang has to be statically built. For this a few modifications inside the cmake file are required.

Edit glslang-master\CMakeLists.txt and add the following two lines after elseif(MSVC) (around Line 185):
set(CMAKE_CXX_FLAGS_RELEASE "/MT")
set(CMAKE_CXX_FLAGS_DEBUG "/MTd")

mkdir build
cd build
cmake ..
glslang.sln
Build Debug|x64
Build Release|x64
PrivacyAmplification will automatically use this build using relative path so no copy is required.

Update RHash:

SHA3-256 is an amazing hash function. The fastest SHA3-256 library I was able to find was RHash. To update it replace the files inside privacyamplification\PrivacyAmplification\sha3\ with the latest version from https://github.com/rhash/RHash - you only need byte_order.h, sha3.c, sha3.h and ustd.h.

Update ThreadPool:
Replace privacyamplification\PrivacyAmplification\ThreadPool.h with the latest version from https://github.com/progschj/ThreadPool

Update mini-yaml:

Replace the Yaml.hpp and Yaml.cpp inside privacyamplification\PrivacyAmplification\yaml\ with the latest version from https://github.com/jimmiebergmann/mini-yaml

Update half_lib (currently unused):

Replace privacyamplification\PrivacyAmplification\half_lib\ half.hpp with the latest version from https://sourceforge.net/projects/half/files/half/

Build:

Build glslang:

```
mkdir "PrivacyAmplification/glslang-master/build"
cd "PrivacyAmplification/glslang-master/build"
cmake ..
PrivacyAmplification/glslang-master/build/glslang.sln
Build Release|x64
```

Build GLSL Shaders:

```
cd "PrivacyAmplification"
./compileGLSL.cmd
```

Build PrivacyAmplification:

```
PrivacyAmplification/PrivacyAmplification.sln
Build Release|x64
```

Build MatrixSeedServerExample:

```
examples/MatrixSeedServerExample/MatrixSeedServerExample.sln
Build Release|x64
```

Build SendKeysExample:

```
examples/SendKeysExample/SendKeysExample.sln
Build Release|x64
```

Build ReceiveAmpOutExample:

```
examples/ReceiveAmpOutExample/ReceiveAmpOutExample.sln
Build Release|x64
```

Build LargeBlocksizeExample:

```
examples/LargeBlocksizeExample/LargeBlocksizeExample.sln
Build Release|x64
```

Docker:

Install NVidia Drivers, Install NVidia Docker, git clone --recursive https://oauth2:3_wtn2uTreSXt9q1ihXV@gitlab.enterpriselab.ch/qkd/gpu/privacyamplification.git and execute ./run_docker.sh. After Docker has started you will be in the Container and see everything running.

How to use tmux:
Linux Docker uses tmux. When starting the container 4 tmux windows will be open. To close them all detach using "Ctrl & B then D” and enter "./run.sh stop” To close a certain window just enter exit. To reattach tmux enter "tmux attach”. To switch tmux window use Ctrl & B then arrow key.

How to stop and exit the docker container:
After detaching tmux just enter exit.

How to start and reattach the docker container:
"docker start container_id && docker attach container_id”

How to SSH into the docker container:
A private OpenSSH and PuTTY key should generate in the same folder as run_docker.sh. The IP is the same as host IP and the Port is 2222.

How to update the container:
It will auto update when execute "./run.sh" if there are no local changes.

How to rebuild the docker image:
"docker rmi image_id” (which you see using "docker images”) then "./run_docker.sh”

<u>Test:</u>

Unit Tests:
Unit Test have near 100% CPU Code Coverage. They work by running a slower CPU implementation and the fast GPU implementation and compare the result. They test multiple sample sizes. Often there are thousands or even millions of asserts per unit test. The test reports it's findings to stdout and assert crashes on failure.

Start Unit tests Vulkan:
```
PrivacyAmplification.exe unitTestCalculateCorrectionFloat
PrivacyAmplification.exe unitTestElementWiseProduct
PrivacyAmplification.exe unitTestBinInt2float
PrivacyAmplification.exe unitTestToBinaryArray
```

Start Unit tests Cuda:
```
PrivacyAmplificationCuda.exe unitTestCalculateCorrectionFloat
PrivacyAmplificationCuda.exe unitTestSetFirstElementToZero
PrivacyAmplificationCuda.exe unitTestBinInt2float
PrivacyAmplificationCuda.exe unitTestToBinaryArray
```

Benchmarks:
```
python speedtestVulkan.py
python speedtestCuda.py
```
Result: See SVGs inside PrivacyAmplification/bin/Release/

Publish Benchmarks:
Requires pip install gists.cli.py3. gists.cli.py3 needs to be configured with a GitHub Token.
```
cd "PrivacyAmplification/bin/Release/"
gists update dfae7685e20cf3f418559f7960e33cfe ?
PrivacyAmpification_RTX_3080_dynamic_seed.svg
gists update dfae7685e20cf3f418559f7960e33cfe ? PrivacyAmpification_RTX_3080_static_seed.svg
gists update dfae7685e20cf3f418559f7960e33cfe ?
PrivacyAmpificationCuda_RTX_3080_dynamic_seed.svg
gists update dfae7685e20cf3f418559f7960e33cfe ?
PrivacyAmpificationCuda_RTX_3080_static_seed.svg
```

<u>FAQ:</u>

Q: How to set which GPU to use?
A: Set the cuda_device_id_to_use option inside config.yaml

Q: How to set PrivacyAmplification parameters:
A: Open config.yaml using a text editor. Carefully read the description inside the config file.

If you have any additional questions feel free to ask.