Compiling Secrecy on UKL

Below, we outline the steps for compiling the secrecy experiment “group_by_join_naive” on UKL (in QEMU on Ubuntu 20.04).

1) Install the packages required for UKL compilation (note: the packages below are for Ubuntu 20.04)

```
$ sudo apt-get update && sudo apt-get install git build-essential flex
bison supermin libelf-dev libssl-dev texinfo libgmp3-dev libmpc-dev
libmpfr-dev qemu-kvm
```

2) Clone the unikernelLinux/ukl repository (note: this repository is private, so the project owners must provide access). If building on Ubuntu, we would recommend using the “ubuntu” branch of this repository. Then, build the UKL dependency files as follows:

```
$ git clone git@github.com:unikernelLinux/ukl.git
$ cd ukl/
$ make all
```

3) Clone the ec528_secrecy repository into the ukl/ folder

```
$ git clone git@github.com:jliagouris/ec528_secrecy.git
```

4) Navigate to the ec528_secrecy/experiments folder and compile (but do not link) the source files for the experiment with the following UKL-specific flags (-mno-red-zone -mcmodel=kernel -fno-pic -no-pie -nostartfiles). Note: at the advice of the secrecy team, we also added #include <libsodium> to the header files for each of these source code files as well as added #include <math.h> to the relational.c file.

```
$ cd ec528_secrecy/experiments
$ gcc  -c  exp_group_by_join_naive.c ../src/comm.c ../src/primitives.c ../src/mpc_tcp.c \
../src/utils.c ../src/party.c ../src/sharing.c ../src/relational.c ../src/baseline.c -std=c99 -O3 \
-Wall -ggdb -mno-red-zone -mcmodel=kernel -fno-pic -no-pie
```

5) Download the libsodium source tarball, untar it, generate the configuration file, run ./configure with UKL-specific flags (-ggdb -mno-red-zone -mcmodel=kernel -fno-pic -no-pie), and then run “make” to build libsodium.

```
$ wget https://download.libsodium.org/libsodium/releases/libsodium-1.0.18-stable.tar.gz
$ tar -xf libsodium-1.0.18-stable.tar.gz
$ cd libsodium-stable
$ ./autogen.sh
$ ./configure --disable-shared CFLAGS='-ggdb -mno-red-zone -mcmodel=kernel -fno-pic
-no-pie'
$ make
```

(note: the libsodium library file is stored in libsodium-stable/src/libsodium/.libs/libsodium.a)

6) Add the following target to the Makefile in the UKL folder

```
#secrecy target
secrecy: gcc-build glibc-build undefined_sys_hack.o
       - rm -rf secrecy.ukl UKL.a
       ld -r -o secrecy.ukl --allow-multiple-definition $(CRT_STARTS) \
              ec528_secrecy/experiments/exp_group_by_join_naive.o \
    ec528_secrecy/experiments/baseline.o ec528_secrecy/experiments/comm.o \
              ec528_secrecy/experiments/mpc_tcp.o
ec528_secrecy/experiments/party.o \
              ec528_secrecy/experiments/primitives.o
ec528_secrecy/experiments/relational.o \
              ec528_secrecy/experiments/sharing.o
ec528_secrecy/experiments/utils.o \
              --start-group
./libsodium-stable/src/libsodium/.libs/libsodium.a \
              --whole-archive $(PTHREAD_LIB) $(MATH_LIB) $(C_LIB)
--no-whole-archive \
              $(SYS_LIBS) --end-group $(CRT_ENDS)
       ar cr UKL.a secrecy.ukl undefined_sys_hack.o
       objcopy --prefix-symbols=ukl_ UKL.a
       objcopy --redefine-syms=redef_sym_names UKL.a
```

7) From the /ukl folder, build the UKL.a file (with the secrecy application) by running make secrecy. Then, compile the Linux bzImage by running make linux-build

```
$ cd ukl/
$ make secrecy
$ make linux-build
```

8) To run in QEMU, edit the “COMMANDLINE” environment variable for the the Makefile in the min-initrd subfolder of ukl by adding the command line arguments for the program (this program expects 3 arguments, for which the last 2 arguments must be powers of 2) after the ‘--’.

   For example:

```
COMMANDLINE = -append "console=ttyS0 root=/dev/sda net.ifnames=0
biosdevname=0 nowatchdog nosmap nosmep mds=off ip
=192.168.19.136:::255.255.255.0::eth0:none -- 0 8 8"
```

9) Run the application in QEMU by running ‘sudo make run’ from the ukl folder

```
$ cd ukl/
$ sudo make run
```