CS5430 Homework 4: Using a TPM

General Instructions. Work alone or with one other person from our class on this assignment. If you do work with somebody, then form a group on CMS and submit a single set of solutions. Any member of a group should be able to, if asked in an oral interview, explain all parts of the submitted solution.

Due: November, 12 2021 11:59pm. No late assignments will be accepted.

Submit your solution using CMS. Prepare your solution as .pdf, as follows:

- Use 10 point or larger font.
- Submit each problem (as a separate file) into the correct CMS submission box for that problem.

Games-R-Us produces a standalone electronic gaming platform called *P1*. This platform comprises (i) a processor with a TPM and (ii) a proprietary operating system called GOS. Many 3rd party developers have written games that run on this platform. So, each Christmas, everyone seems to want to buy a new P1!

The games that run on P1 do not invoke TPM instructions directly. Instead, they invoke a GOS system call, which in turn invokes the appropriate TPM instruction. GOS runs in system mode, and games run in user mode. Recall (according to page 306 of the written course notes) TPM instructions cause a trap unless invoked in system mode.

The TPM in P1 resembles the hardware that is outlined in section 11.3 of the written course notes with the following specifics.

- Measurement registers: mr_0 , mr_1, …, mr_7
- Sealing key registers: skr_0 , skr_1 , … , skr_7
- Quoting key registers: $qkr_{id}, qkr_1, \ldots, qkr_7$
- Unbinding key registers: ukr_0 , ukr_1 , … , ukr_7

Sealing in this TPM uses AES-256 shared key encryption; quoting and unbinding use RSA with 2048-bit keys.

A new TPM --- called *TPMplus* --- was designed by *Games-R-Us,* to be the basis for an improved gaming platform to be called *P1plus*. TPMplus was designed to provide twice the number of measurement, sealing key, quoting key, and unwinding key registers. However, the first silicon-fabrication run of the new TPMplus design yielded chips that have a problem:

- none of the sealing key registers is reliable, and thus, none of these registers is usable.
- whether in system mode or in user mode, an attempt to execute a TPMplus instruction that references a sealing key register causes a trap, which transfers control to the GOS trap handler.

You have been hired by *Games-R-Us* and asked to help determine what, if anything, can be done to have the 3rd party legacy applications that currently run on the P1 platform also run on the P1plus gaming platform.

Problem 1 [at most 1 page] For each of the TPM instructions listed in section 11.3, state whether execution of that instruction is expected to work correctly on a TPMplus. For those TPM instructions that are unlikely to work correctly on a TPMplus, explain why.

Problem 2 [at most 2 pages] *Games-R-Us* management has decided that it is unrealistic to ask 3rd party legacy developers to alter their programs for running on the (flawed) TPMplus. Instead, *Games-R-Us* management is hoping to modify GOS in ways that mask the problems with the (flawed) TPMplus, so applications run unchanged. But changes to GOS that would downgrade the security of P1plus are considered unacceptable by *Games-R-Us* management.

For each of the problematic instructions you listed in Problem 1, either (i) explain why GOS cannot be modified to mask the flaws in TPMplus or (ii) explain what modifications to GOS would allow GOS and 3rd party legacy applications to continue invoking that problematic TPM instruction.