Chapter 8

Select implementation considerations

Here we briefly list some of the potential challenges and some other aspects that are still left open for further discussion and research.

SPI Flash emulation challenges

One anticipated complication for emulation of the SPI flash by the trusted stick is that the processor (chipset) expects the specific timings to be met by the SPI chip when reading firmware, so it's unlikely one could use a general-purpose uC on the stick to emulate the flash chip. Also the timing requirements make it unlikely that a regular SD storage card will work for us here.[1] Rather, we need a real SPI flash chip located on the trusted stick, or better: an FPGA-based implementation.[2]

Also it does not seem trivial to use the same one SPI chip to both serve the firmware (i.e. ME, BIOS, other) to the host processor, and at the same time to also act as a flash provider to the EC, and optionally also to the internal disk. The primary reason for this might be lack of a good multiplexing mechanism built into the SPI protocol. This seems, however, merely a technical complication that, in the worst case, could be resolved by having the Trusted Stick exposing two

[1] Which otherwise sounds like a great solution, at least for prototyping, as most of these cards should be implementing the simple SPI protocol.

[2] The reason to use an FPGA-based implementation of an SPI flash is transparency, required to assure that our Trusted Stick indeed implements read-only protection for certain parts of the flash, as well as reliable encryption for other partitions, as discussed earlier in the paper.