it turned out that e.g. Intel ME refused to run having only read-only access to its flash partition, then we might need to encrypt the flash partitions on the Trusted Stick holding this early boot firmware. More on this at the end of the paper.

An alert user might be wondering what a TPM device is doing on a stateless laptop? After all the TPM has its own non-volatile memory, doesn't it? Interestingly on the recent Intel platforms the TPM has been integrated into the processor package (it's in fact an application running on the ME processor), and so it uses the system's SPI flash memory as its own non-volatile storage. Of course everything that is written there is encrypted with a key that is tamper-proof protected inside the processor, so the mere fact the attacker is able to read the SPI flash content with an external programmer does not compromise safety of this TPM's storage. While it hasn't been confirmed experimentally if such a processor-internal TPM would work with a read-only storage exposed by the Trusted Stick, it seems plausible to expect it should[2]. Of course the user would be expected to let the TPM write its generated keys during the platform initialization, by operating the read-protect switch on the Trusted Stick.

The variant with internal trusted disk

As already discussed earlier, assuming a trusted, open implementation of an internal hard disk was available, then the stick would not need to act as (fast) storage. It would only have to provide the decryption key to the (trusted) internal disk device.[3]

The primary benefit in this case would be the simplification of the stick: no need to fit high-capacity, high-performance flash memory. Depending on the application this could be an important benefit.

Self-destruct

Optionally, at least for some groups of users, it might be desirable for the Trusted Stick to implement quick and reliable wiping of its content, especially of the user partition.[4] This should be easily implemented by securely erasing just the

[2] And in case it didn't work with a read-only flash, we might still be able to use it with an encrypted writeable flash, as discussed later in the paper

[3] Potentially it might also be providing the /boot partition, although the benefit of this is unclear.

[4] Although, there might be scenarios extending this requirement also for other partitions, i.e. these holding the firmware and system image.