User partition encryption considerations

It seems tempting to delegate the user partition encryption to the host – after all it runs the user approved trusted code from the stick's read-only partition, while at the same time this simplifies the construction of the stick significantly.

Unfortunately, running the encryption on the host processor we're exposing it to potential malicious interference from the ME processor. The ME can e.g. steal the encryption key from the host registers or memory pages and then try to leak it through some of the user networking activity, although this might be very difficult in practice as discussed earlier in the paper. What the ME can do, however, and very simply, is to store some of the leaked user sensitive information (such as the email private keys) on the user private partition *without* encrypting them with the user key, but rather with some other key. This would then look like random garbage for the user, if they ever decided to examine the sectors on the partition. But for the attacker who (physically) obtains access to the user stick this might be immediately readable.

On the other hand, if it was the Trusted Stick that performed the encryption, then there should be no way for the hypothetical ME rootkit to write anything onto the user partition bypassing the forced encryption with the user key.

Temper-resistance considerations

The use of tamper-resistance technology is often thought as a beneficial means to improve physical security of an endpoint device. Care must be applied however as to whether this does not compromise the ultimate trustworthiness of the product.

In the author's opinion it is unacceptable for any *code*, that the user is forced to entrust their digital life to, to be tamper-proof-protected if that results in an inability for the user to dump and analyze the code that runs on the device at any time the user feels a need to do that.[4]

Thus a temper-proof mechanism might only be acceptable for the actual (small) persistent memory which holds the bits of the user keys, and for nothing more, particularly not for the memory which holds the firmware for the device. Also, any tamper-proof protection on volatile memory (RAM) is not necessary, as such protection only makes sense if the threat model assumes the legitimate user to

[4] And it is completely irrelevant whether the user would, in practice, be willing or capable to do that or not – it's a matter of having an *opportunity* to do that. This is very similar to guarantees of civil liberties, such as free speech.