size (which is an important factor given the stick is assumed to be carried by the user with themselves all the time, perhaps in a form of a necklace, or maybe even a ring in the future).

2. Even more importantly: it significantly enlarges the attack surface on this trusted device. Admittedly the uC used for networking proxy implementation might be a physically different one than the chip used for SPI firmware exposure, although this would now complicate the host-stick interface, in addition to further increasing the cost and size.

However, just as we discussed the use of a stateless internal disk (which runs a trusted firmware from the stick), we could similarly envision a simple networking proxy implemented using a stateless (i.e. flash-less) uC, which would then connect to a traditional WiFi card. The WiFi, however, would *not* be directly connected to the host CPU.

Incidentally we have already outlined the need for a stateless uC on the laptop – this is to implement the Embedded Controller. It seems thus logical to use this same uC for both realization of the EC as well as for the (trusted) networking proxy.

Obviously it would take time to write firmware implementing the envisioned proxy, and before this one is ready, a temporary solution could be to use an external, USB-connected or Ethernet-connected network proxy (similar in nature to e.g. [13]).

Scenario 2: Tor-ified or VPN-ed open Internet

Let's now consider the traditional scenario in which the user wants to interact with any computer on the Internet, whether trusted or not.

In this scenario we would also like to use the previously discussed networking tunneling proxy. Of course at some point the tunnel would need to be terminated and the user connection will now be visible to some 3rd party Internet infrastructure, including the final 3rd party server (e.g. a cat-photo-serving website the user might be addicted to). The termination of the tunnel would take place at a VPN service provider (which we assume to be a trusted service provider for the user), or at a Tor exit node (which itself is not assumed to be trusted, but the Tor network, as a whole, should be in that case).

Now, assuming the malware has modified the content of the user-generated packets high enough (OSI-layer wise), such as e.g. modified some of the HTTP(s)