possible to use it to exfiltrate the stolen data[4]. Of course, this would result in a "black-hole" use model – the air-gapped system can only accept files from the outside world, but never give anything back to the universe – again, possibly a sub-optimal use of computer technology...

Scenario 1: A closed network of trusted peers

Now, let's consider a closed network of trusted peers who would like to communicate securely with each other, also exchanging files.[5] Of course the humankind has researched this problem extensively over the last couple of decades, which resulted in an abundance of cryptographic protocols allowing to build secure tunnels over insecure networks.

However, assuming a rootkit running in the ME or SMM, we're suddenly facing a significantly more difficult challenge. This is because the ME might be now piggybacking stolen information (such as the session keys for the crypto tunnels we're trying to build) on the existing network packets, allowing an adversary – who e.g. controls the user's ISP – to receive them on a plate.

In order to prevent this from happening we need to move the actual networking device away from the jurisdiction of the ME and the host processor. It seems convenient, at first thought, to place the networking device on the Trusted Stick. Indeed, if the trusted module was implemented as a USB-pluggable device then it would be able to provide emulated Ethernet device to the host. The Trusted Stick would then perform simple tunneling to establish the virtual trusted network with other peers (hopefully using also similarly designed laptops). This way, even if a hypothetical ME rootkit was trying to leak some information over networking, this would get encapsulated into the encrypted tunnel, which only the trusted peers were able to see.[6]

Implementing Ethernet-emulation and networking proxy on the Trusted Stick has several disadvantages though:

1. It complicates the Trusted Stick design, increasing its cost, as well as its

[4] Although one should remember the DVD-R driver will likely be fitted with its own uC featuring its own flash memory, which might be a good candidate for malware to store stolen secrets to.

[5] Again, this means, by definition, that any of these "trusted peers" is able to compromise the whole network.

[6] Admittedly, as several reviewers noted, the rootkit might try to leak the stolen keys by interfering with the timings of packet transmissions, or using some other sophisticated side-channel attack...