

1) Which threats to mobile devices does UDPI protect against?

UDPI protects against the following threats in particular:

- a) Protection of system boot integrity
- b) Protection of the operating system integrity
- c) Protection and isolation of security critical applications (e.g. payment applications)
- d) Protection of access control information
- e) Resistance to code vulnerability exploits, depending on the implementation
- f) Digital rights concerns, although UDPI is not a DRM-based protection

2) Can UDPI's technology be hacked through software or patches available on the web?

No, with UDPI one cannot change the code nor download software or patches from the web to hack the system. The UDPI code is unique for each device.

3) Does UDPI use a non-volatile or one-time programmable memory for storing device keys?

No, because the UDPI key is inherent in the device.

4) Can UDPI be used with other technologies such as algorithms or authentication processes?

Yes, UDPI can be used as the core, running algorithms or authenticated processes required for other technologies. UDPI itself is non-algorithmic and is compatible with many other technologies.

5) Is there a certain type of flash or memory required for UDPI's technology?

No. For UDPI all that is required is a regular memory. There is no need for an authenticated flash or any other specialized memory product.

6) Can UDPI prevent a hack involving two sets of flash, with a valid flash verifying code at the beginning of the process, and then a "hack flash" executing code once the seal of approval has been provided?

Yes, UDPI can prevent such a hack. The above scenario is impossible with UDPI since the code is uniquely encoded for each processor and in some embodiments of UDPI, the address lines may be accessed in a totally random order. With UDPI, even if an attacker accesses the circuitry, they would have to know the UDPI key for the memory address to decode data. The beauty of UDPI is that the key for an address is different from one device to another, even for the same address.