Mobile software security

Chris Mitchell

Convergence is a word that promises much for the PC and mobile telecommunications industries. However, it also a word that should worry us all because of the potential security implications.

We have all become used to the advantages and disadvantages of the openness of the PC platform. We use software applications from a huge number of different sources, without even thinking about it. We have an enormous amount of control over our own PCs, including the ability to choose exactly which software we run. This has huge advantages in terms of flexibility, and also helps create a large and vibrant market in PC applications.

However, this openness also brings with it major problems. Malicious software of many kinds is common, and is a constant threat to system integrity. This is to a large extent true because of the very openness that we value. Not only do we have the ability to run software of possible unknown origin, but even our 'well trusted' applications, e.g. web browsers, allow us, even encourage us, to permit web sites to download and execute applets on our PCs. In most cases we, the PC users, have no way of knowing whether we should trust the originators of such software, and so we simply accept the software in order to get the functionality we want. No wonder we have so many security problems!

The traditional mobile phone is rather a different beast. In the past, the software within such devices has been fixed, and the ability to reprogram such phones has been very limited. For example, we might be able to change the ring tone, but nothing of any real importance.

Convergence means that mobile phones will soon have functions we expect of a PDA and more, whether we like it or not. This, in turn, suggests that our phone will gradually come to resemble a PC. Of course, this brings enormous potential benefits. We will be able to customise our phones in all kinds of ways, including downloading applications from a large variety of different sources.

This initially seems very appealing; however, do we really want a phone which will be prone to virus infection and which will need rebooting at regular intervals? Indeed, which is more important – flexibility and openness or reliability? Moreover, once the flexibility arrives, and it appears to be hard to resist, how will we get back the reliability and simplicity we have come to expect? Of course, one solution would be to allow our mobile phone manufacturer to decide for us which code should be executed on our device. However, the network operators, and also many users, are unlikely to be happy with such a situation. Equally, the network operator could manage all such issues, but this would be unlikely to be welcomed by the manufacturers.

The MExE (Mobile Station Application Execution Environment) initiative has been designed to help reduce the potential chaos associated with downloading new software into our mobile devices. The aim of MExE is to provide a secure standardised environment for executing applications. That is, MExE will automatically control which applications are permitted to execute on your mobile device. The integrity and origin of pieces of code are guaranteed by digital signatures, the validity of which are checked using 'root' public keys embedded in the phone (a bit like the root public key embedded in your PC web browser).

However, problems remain with MExE. It remains to be seen how widely adopted MExE will become. Moreover, the effectiveness of MExE in preventing the spread of malicious code also remains to be tested.

One possible source of problems is the MExE security architecture itself. MExE tries to keep both phone manufacturers and operators happy by providing two separate 'environments' on the phone for code approved by manufacturers and operators respectively. A third environment, for third party approved code, also exists. The precise privileges associated with the three domains varies slightly. Whether such an architecture is sufficiently flexible to meet user needs, and yet sufficiently restrictive to prevent large scale malicious code problems, remains to be seen.

What is clear is that, in the future, there will be a constant tension in many different application domains between requirements for robustness and reliability and requirements for openness and flexibility. Exactly similar problems arise not just for mobile phones but in completely different areas such as motor vehicles, where there are growing possibilities for flexible reprogramming of engine management systems. Not only do issues of reliability arise, but major regulatory problems are also looming. The 'soft world' has plenty of security surprises for us yet!