Algorithms and Kerckhoffs' principles

Chris Mitchell

In 1883 Auguste Kerckhoffs published a two-part article entitled *Military cryptography* which has been enormously influential in the development and understanding of cryptography. For the interested reader, Fabien Petitcolas has done a wonderful job of making the article accessible – see http://www.cl.cam.ac.uk/~fapp2/ for more details.

Probably the most widely quoted aspect of Kerckhoffs' article is a series of desiderata, now known as Kerckhoffs' principles. These principles give the basis for assessing a cryptographic system, and are still widely accepted today. Of these, probably the most important is one which implies that, in assessing the security of a scheme, it is necessary to assume that the method used to encipher data is known to the opponent, and that security must lie in the choice of key and not in the secrecy of the algorithm. As Petitcolas puts it on his web page, this does not necessarily imply that the method must be public, but should be considered as public during its creation and assessment.

The reason for this is clear – any automatic system which is widely deployed has a significant probability of falling into the hands of the 'enemy', who can then reverse engineer the implementation to find out how it works. All that the legitimate users can safely rely on is the key remaining secret, and hence the design must be such that finding the key is infeasible even if the design of the system is public.

Recently, the meaning of this principle has on various occasions been subtly modified to suggest that all algorithms should be made public. Of course, there are advantages with making algorithms public – not least it means that an algorithm is likely to be well-scrutinised for weaknesses by the global cryptographic community, who like nothing better than breaking new schemes. However, there are also clear advantages with keeping an algorithm secret – it is hard to cryptanalyse something when you don't even know how it works, and why make life easier for your opponent? Keeping an algorithm secret remains standard practice for government and military use.

One well-publicised example of this misuse of Kerckhoffs' principle relates to the GSM encryption algorithm known as A5/1. This algorithm, which is used to encrypt telephone conversations between a mobile handset and a 'base station', has remained a secret since its design in the 1980s, despite the fact that it is built into every mobile handset. Recently the design has been made public via the web. Still more recently, A5/1 has been successfully attacked by Biryukov, Shamir and Wagner using some rather clever and novel cryptanalytic techniques. I should say at this point that it is not clear how relevant this attack is in practice, since, using the jargon of cryptography, the attack needs a certain amount of *known plaintext*, i.e. a number of bits of the digitised voice signal making up a telephone call. To confuse matters a little, a so-called *alleged A5* was made public several years ago, although it has important differences from, and is much weaker than, the real A5/1.

The designers of A5/1 have been accused of ignoring Kerckhoffs' principle in not publishing the A5/1 algorithm. Of course, it is impossible to know how the A5/1 design was assessed, since the design process has never been made public. However, the mere fact that the algorithm was not made public is in itself clearly not in breach of the principle.

To conclude, it is interesting to ask what the A5/1 designers ought to have done. First observe that strong public domain stream ciphers are very hard to find. Recent calls for proposals for stream ciphers to include in an ISO/IEC encryption standard (ISO/IEC 18033) have meet with no responses at all! By contrast, the world is knee-deep in good block ciphers - the next draft of the ISO/IEC encryption standard is expected to contain eight different block ciphers!

The GSM system architects were thus essentially forced to design an algorithm for themselves. They were faced with the additional problem of choosing an algorithm which would be simple and cost effective to implement in all mobile handsets using 1980s technology. The solution they came up with, namely A5/1, probably represents the state of the art at the time. What might have happened had it been published along with the rest of the GSM specifications? Well, my guess is that it would have been broken much earlier, but probably too late to have prevented the wide scale deployment of GSM equipment containing the algorithm – just the situation the designers sought to avoid. Keeping the algorithm secret probably bought ten years of security for GSM encryption, probably as long as anyone could have expected.

Of course, it could be argued that we are now in the 'worst case' scenario – world-wide deployment of GSM based on an insecure stream cipher. However, even now the algorithm is not trivially easy to break – it is difficult to assess how easy it might be to get hold of the necessary known plaintext to make an attack. My guess is that A5/1 is still good enough to prevent casual eavesdropping on the radio link, which was probably always the main objective of GSM encryption. As for Kerckhoffs' principle, I think I will stick with the original version.