



DeadMan's Handle and Cryptography

Introduction

This paper describes the relationship between DeadMan's Handle (DMH) and the use of cryptographic systems on notebooks. It explains why DeadMan's Handle can be a valuable addition to the use of cryptography in enhancing notebook security. This document attempts to give a non-technical overview; most especially, it is not a primer in cryptography or the operation of DMH.

Cryptography

Cryptography is a growing area. As a technology, it has migrated from the almost exclusive domain of the military to being endemic in civilian applications. Whilst the first commercial uses were in banking, cryptographic systems have moved into the mainstream of computing. They are, for example, heavily used in the communications links in the World Wide Web – especially when purchasing is involved.

One area where cryptography has grown recently is in the protection of information on computers. This can be implemented in any number of ways, for example:

1. The information may be encrypted by the disk operating system.
2. There may be special 'safes', or 'logical drives': any files placed in these will be encrypted.
3. Programs may specifically encrypt certain designated files.

In all three cases the systems often require the use of passwords, which are used to control the cryptographic keys. In some cases the password may be tied to the sign-on password.

The security aim is that if the notebook is lost then any thief will be unable to read the encrypted information, as the attacker will not know the password. Without knowing the password, the information looks like random numbers



and may not even be obviously accessible (for example the 'safe' cannot be opened).

Clearly in many cases this will be a highly effective means of security. Should the notebook go missing the encrypted files will be normally impossible to read, especially as modern systems use extremely powerful cryptographic routines.

Limitations of Cryptography

Unfortunately there are limitations to the use of cryptography and it can also have unexpected side-effects. Perhaps the first limitation is that many cryptographic systems assume the machine is in a controlled environment. Windows-based encryption is generally based on the idea that the machine is protected by the logon password, which directs any specified encryption. However if the notebook has gone missing then it is a very simple procedure to reset the Administrator's account password: this account may then allow the thief to 'recover' the encrypted files as the 'Data Recovery Agent' (see reference [1] for comments on this).

However, cryptographic systems can fail for many other reasons (see references [2] and [3] for a background discussion of this area, these papers also have a number of references which are useful). One general weakness is poor usage of the system by individuals: easily guessed passwords to the cryptographic system are chosen – or even written down on a sticker and stuck on the notebook. Or people use the same password across different systems. The password then becomes as safe as the weakest system that is using it (which will often be something like a web browser which has been told to remember the password).

Another level of problems lies with organisational misuse of encryption systems. As these systems get larger, they generally meet what is known as a 'key management' issue. Put simply, different people want to read the same encrypted files. Therefore these people need access to the key that lets them read the files. Yet the key cannot be known to everyone; what is



worse is that the organisation needs different keys for different groups of people, who want to share information whilst excluding others. With one or two people key management is easy, with 10,000 it becomes a nightmare.

Although methods have been invented to help deal with the problem (such as one called 'public key infrastructure') they have tended to mitigate the problem rather than solve it. They have also introduced complexities of their own. This has led to organisations effectively misusing cryptographic keys as managing them appropriately has become too complex. The simplest example is that everyone in the organisation really does get the same key.

All these issues tend to lead to possible compromise of the cryptographic defences on the notebook. If an opportunistic thief of the laptop sees a sticky label on the side with 'axybbb' written on it he or she may well make certain deductions.

There is another concern with cryptography: it marks a notebook as interesting. The fact that effort has been taken to encrypt information would imply that there is something valuable on the machine. Once this has been detected, the thief may – instead of selling it at the nearest car boot sale – take it to a more experienced contact for analysis. The possible results might be unpleasant.

The Use of DeadMan's Handle

DeadMan's Handle is envisaged as an addition to cryptography, not a replacement for it. It provides an extra level of protection – what is known as 'security in depth' – always regarded as a good security model to follow.

DMH operates on rather different premises from cryptography; most importantly it is customised to the individual, who decides what is confidential and what is not, and what level of security is required. Thus the user designates the secure folder and any extra files, and then decides the keycode or password.



Here the first benefit of DMH comes in. The keycode can just be a single number, or the password one letter. Very simple passwords are acceptable because the thief is going to get a very limited number of tries under normal DMH settings: fail those tries and the information is deleted.

The next benefit is that DMH quietly deletes the files, concealing their names and timestamps. It then deletes itself. The thief is left with an ordinary looking machine, which still works, that might have some trivial and unimportant files on it. There is no indication that there ever was anything interesting on the machine – so off it goes to the car boot sale.

The third benefit is that should the machine come into the possession of a scavenger, there is nothing to scavenge. The files have been carefully damaged so that they are irretrievable even using a file recovery utility. The filenames may look somewhat unusual, but do not display any obvious points of interest.

Lastly, under certain cases the use of cryptography can accelerate the operation of DMH: the two systems can operate in tandem. See [1] for more information on this.

Thus with both systems in place the notebook is protected at two levels: the cryptographic level and the DMH level. It means that the attacker has to become very sophisticated to have a chance of recovering confidential information from the machine. This raising of the barrier is the fundamental concept of security, and we would recommend both approaches to companies with valuable information on their machines.



References

- [1] *DeadMan's Handle: Tips for Use*, DeadMan's Handle Ltd,
<http://www.deadmanshandle.com>.
- [2] *Lessons Learned in Implementing and Deploying Crypto Software*,
Peter Gutmann, <http://www.cs.auckland.ac.nz/~pgut001/>.
- [3] *Why Cryptosystems Fail*, Ross Anderson,
<http://www.cl.cam.ac.uk/users/rja14/>.