RISOFTDEV inc. black key technologies

Vincent L Gilbert MCSE MCSA MCP

Abstract

In this paper we will provide a brief overview of the various technologies in the field of crypto logic which have been developed at RISOFTDEV inc., and the types of commercial products that might be developed from those technologies. R.I. Keys is a benign key creation system, designed to be used with any software. Keyfiles can be created locally by using the keyfile creation tool, or by using online versions of the creation tool. Keyfiles now can be set to any length and so can be used with any encryption algorithm.

Other systems

Other Black key generators are hardware based, and rely on a KEK or Key Encrypting Key. The KEK is used to encrypt the key for transportation. This works in hardware based systems where the “black box” is relatively secure, but in a home computer environment it is not suitable. The question is “where do you store the KEK?” The answer was in eliminating it.

Our technology

R.I. Keys represents a leap forward in Black Key technology. Instead of a master KEK, environmental and other factors create a KEP, or Key Extraction Pattern. This KEP is used to extract the key from a Keyfield. (See ‘Keyfield’ above) Because the key itself is not visible, any attempts to back engineer the algorithm will arduous at best. However we acknowledge that relying on this fact is not in keeping with accepted cryptographic practices. Kerckhoffs principle — "only secrecy of the key provides security", or in Shannon's maxim, "the enemy knows the system" simply states that any encryption system has to accept that given enough time and access to the system, a dedicated group will be able to discern its mechanics. With this in mind, most encryption algorithms are made public and depend on mathematical complexity . In this case we seem to be, and are in fact

violating this principle. To do so we are introducing what we are calling **Extensible Obfuscated Algorithm Encryption** technology. In **EOAE** the algorithm which creates the key is hidden, but it is accepted that given access to enough keys a dedicated effort will be able to discern it according to the foregoing principles. This is expressed as the value T. However elements of the algorithm are extensible, that is they can be altered simply, and without changing the fundamental mechanics of the algorithm. Since the algorithm is meant to be used by a known group, the updated version of the algorithm can be made available to this group within the time T, or upon a determination that the algorithm is compromised. The existing key is exported and a new key using the updated algorithm is sent to the user. Anyone attempting to crack it is forced to start over. The design allows for a nearly limitless number of patterns. This technology will NEVER be obsolete. From a business model perspective, it is ideal as the provider has a service that they can offer in perpetuity.

Problems associated with key creation

True random number generation is not possible with the computer, yet it is necessary for generating secure non repeating sequences. R.I. Keys reduces this problem by introducing the **Master Keyfile.** The master keyfile magnifies entropy by increasing the number of characters generated, and then randomly selecting the string length needed from within this extended group. The use of a master keyfile greatly decreases the chance that disparate installations will duplicate a given Keyfield.

Description

If you were to view a typical key file in a text editor, you would see something like this.

```
---------------------------------------
r.i. encryption ver 3.0
---------------------------------------
(^)Q^&TER_(Q*[fljQBW$@!*
(Q*[fljQBW$@!*$_!&851I=]
y165vqkWJ@!*BDV[=OQ2457U
5vqkWJ@!*BDV[=OQ2457UISG
GC=9QUBO[ojv]oweri]0webf
ljQBW--97qwt-ey_(Q*[fljQ
138DC5E39C9C663F1362DBB03648E1CCF277A129
test1
---------------------------------------
```

The key file can be divided into the following sections

Header

---
r.i. encryption ver 2.2

The header contains version information for this keyfile

Keyfield

(^)Q^&TER_(Q*[fljQBW$@!*
(Q*[fljQBW$@!*$_!&851I=]
y165vqkWJ@!*BDV[=OQ2457U
5vqkWJ@!*BDV[=OQ2457UISG
GC=9QUBO[ojv]oweri]0webf
ljQBW--97qwt-ey_(Q*[fljQ

Contained in this field are the base characters which make up this key. The characters are not contiguous, but are made up from Geometric patterns which do not recur from key to key.

(^) **Q^&TER_(** Q*[fljQBW$@!*
(Q*[fljQBW$@!*$_!&851I=]
y 165vqkWJ@!*BDV **[=OQ2457U**
5 vqkWJ@!*BDV[=OQ2457UISG
G C=9QUBO[ojv]oweri]0webf
ljQB W--97qwt-ey_(Q*[fljQ

Hash

138DC5E39C9C663F1362DBB03648E1CCF277A129

The hash locks the keyfile to its owner, and to the originating computer. Special export functions are available that allows the keyfile to be used from other locations.

Owner

test1

The owner of the keyfile

Footer

Summary

Our obfuscated key technology greatly enhances the effectiveness of any encryption scheme. It can be integrated quickly and simply into any third party software, making it the perfect value added upgrade for any company wishing to offer its customers real protection. In many cases it can replace x.509 certificates, overcoming many of the problems which have plagued PKI since its widespread implementation.