Technical White Paper

SECURITY SOFTWARE

Contents

Introduction

In the SafeIT product portfolio, three main areas of security technologies are included.

- File Encryption
- E-mail Encryption
- File Shredding

In the different SafeIT-products the above functionality's are being used differently depending on product packaging. However, this paper aims to give insight to the general implementations of technology in the complete SafeIT product portfolio.

The purpose of this chapter is to provide a basic presentation of the SafeIT Encryption Solutions. This document is a summary that aims to highlight certain important characteristics of technical functions. It assumes that the reader is familiar with the concepts of cryptography and both symmetric and asymmetric encryption.

For those interested in further knowledge within the field of cryptography, the book "Applied Cryptography" by Bruce Schneier provides an introduction to cryptographic methods and an extensive bibliography. This publication is highly recommended.

This document does not define the legal scope or extent of the intellectual property comprising the SafeIT encryption system.

SafeIT File Encryption

The SafeIT product portfolio contains two different versions of file encryption.

1. SafeIT File Encryption
   Encryption of an individual selected file or folder stored in your normal data file structure (NTFS/FAT).

2. SafeIT Secure Disk which is a Virtual Encrypted Drive.
   Creation of a new virtual drive where all information automatically is encrypted.

The first alternative offers:

All symmetric encryption algorithms listed below.
Possibility of creating self extracting encrypted files (*.exe).
Multiple encryption, i.e. one file can be encrypted several times with different keys and algorithm combinations.

Symmetric encryption algorithms:

- New AES-standard Rijndael[i] (key length: 256 bits)
- Blowfish[ii] (key length: 448 bits)
- Twofish[iii] (key length: 256 bits)
- SafeIT-algorithm (key length: 480 bits)

Pseudo Random Number Generator (PRNG):

- Isaac[iv]

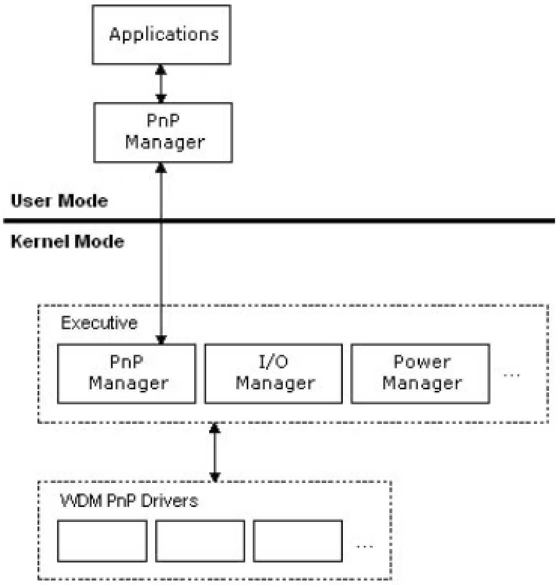
Hash-algorithm:

- MD5[v]

Compression algorithm:

- ZLib, RFC1950-RFC1952[vi]

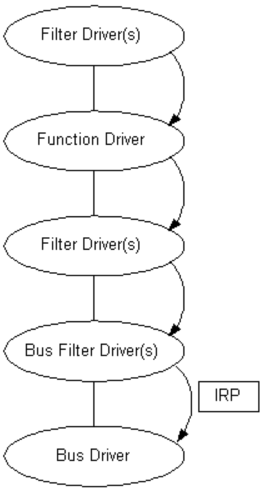
The Second alternative offers:



Creating a SafeIT virtual encrypted disk is done according to the following general approach.

A function driver design is used. The program emulates a hard disk but to the system it appears to be a system device, not a storage device.

A storage driver handles I/O requests from user applications or higher-level drivers and sending them to the underlying storage bus driver in its WDM stack.



The use of a WDM function driver leads to a hardware and platform independent driver. When the driver is not a storage device the function driver has no possibility to send the information down to a bus driver and a hard disk. This is done by creating a file from the driver using the internal function in Windows.

The advantage of this design is that the driver is easy to migrate to another Windows environment without any possibilities of conflicts with other hardware or drivers.

Symmetric encryption algorithms:

- New AES-standard Rijndael (key length: 256 bits)
- Blowfish (key length: 448 bits)

Asymmetric encryption algorithm:

- Diffie-Hellman Key Exchange Protocol[vii]

Hash-algorithm:

- SHA-1[viii]

Pseudo Random Number Generator (PRNG):

- Isaac[ix]

SafeIT E-mail Encryption

SafeIT establishes a secure encrypted e-mail connection by using the Diffie-Hellman Key Exchange Protocol. Once the first encryption key is calculated, the symmetric algorithm is used in the program together with an automatic key exchange system.

First encryption key exchange
Diffie-Hellman key exchange method using 2048-bit long encryption key.

Symmetrical Encryption Algorithms

- New AES-standard Rijndael[x] (key length: 256 bits)
- Blowfish[xi] (key length: 448 bits)
- Twofish[xii] (key length: 256 bits)
- SafeIT-algorithm (key length: 480 bits)

Pseudo Random Number Generator (PRNG)

- Isaac[xiii]

Automatic encryption key exchange
Encryption keys are automatically exchanged each communication round. New encryption keys are transmitted encrypted between parties.

Encryption key usage
One unique current encryption key per counterpart.

General Capacity
Depending on the configuration of your computer.
3-13 Mbits/s on a Pentium II, 333 MHZ.
Never a setback for the user.

Number of possible secure connections
Indefinite

Possible plaintext
All digital files can be encrypted, e.g. text, pictures, program files etc.

Compression algorithm and its strength
ZLib is used for compression. How much the information can be compressed depends of which type of information the file contains. The compression can be up to 98%. Normally 50-60%.

IV
The IV is saved encrypted with the current key with IV=0 (IV is reset when encrypting).

The SafeIT™ Algorithm

The SafeIT Encryption Algorithm is a secret key stream block cipher. The algorithm consists of linear and non linear operations which are performed in rounds. XOR operations are included in this process.

The algorithm compresses all information that is being encrypted. The algorithm uses an Initializing Vector that randomly starts off the encryption process.

The algorithm contains encryption key dependent S-boxes.

Characteristics of encrypted information

Information encrypted by the SafeIT-algorithm is always compressed to a maximum. There are no known patterns, linearity or symmetries to be found in the encrypted information.

If one bit is changed in the plaintext, the encrypted information is completely changed.
If one bit in the encryption key is changed, the encrypted information is completely changed.

The exact same encryption key used together with the exact same plaintext will generate different encrypted information depending on the Initializing Vector.

Pseudo random number generator

Each encryption key is generated by a pseudo random number generator. The private key uses the pseudo random number generator 2048 times to generate the key used. Each symmetrical key is generated by running the pseudo number generator 120 times to produce a hexadecimal character between 0-15. Each hexadecimal character then consists of 4 bits and thus the 480 bit long encryption key is created. The Isaac PRNG is used in this context.

Key Exchange

The SafeIT encryption system continuously exchanges the 480-bit long encryption key. On average a key is used between two counterparts to encrypt and decrypt two documents.

Key Management

The key exchange process is fully automatic and requires no key management by the user.

Key Transmission

After successful start of exchanging keys, new symmetric encryption keys are transmitted between the counterparts encrypted together with the other encrypted information.

Authentication

Since each encryption key is unique and secret to anyone besides the correct counterpart, the identity of the sender is confirmed each time a correct decryption is performed. The only way to get hold of a unique encryption key used by two counterparts, for one round of communication, is to steal or hack one of the computers that are being used when sending the encrypted message.

Data integrity-electronic seal

Since the encrypted information is dependent on each bit in the plaintext and the encryption key, a correct decryption cannot be performed if any information is changed during the transmission. Correct decryption guarantees data integrity.

Man in the middle attack

When using asymmetric encryption to start a secure connection, you are vulnerable for a “man-in-the-middle” attack. If someone is eavesdropping on you and scans every e-mail sent and received, and completely controls your communication, this attack can be done. Before a secure connection is established, someone can place himself in between the two counterparts and simulate a secure connection with each one of the two counterparts. This is difficult to perform but theoretically possible.
To control this, SafeIT allows you to view encrypted e-mails. If two counterparts compare the same encrypted e-mail the content should be exactly the same. If so, a “man-in-the-middle” attack has not been performed and automatic security has been established correctly. This means that the secure connection cannot be attacked. It is also possible to compare the two counterparts’ public keys used when establishing the secure connection. If these keys are the same on both sides when they are compared, a “man-in-the-middle” attack has not been performed.

SafeIT File Shredding

The SafeIT Shredding Technology is based on well known standards and approved algorithms for complete data removal.

Overwriting Algorithms

- HMG Infosec Standard 5, The Baseline Standard. The algorithm using 1 pass of overwriting.
- HMG Infosec Standard 5, The Enhanced Standard. The algorithm using 3 passes of overwriting.
- Peter Gutmann's algorithm. The algorithm using 35 passes of overwriting.
- U.S.Department of Defense Sanitizing (DOD 5220.22-M). The algorithm using 3 passes of overwriting.
- Bruce Schneier's algorithm. The algorithm using 7 passes of overwriting.
- Navy Staff Office Publication (NAVSO P-5239-26) for RLL. The algorithm using 3 passes of overwriting.
- The National Computer Security Center (NCSC-TG-025). The algorithm using 4 passes of overwriting.
- Air Force System Security Instruction 5020. The algorithm using 4 passes of overwriting.
- US Army AR380-19. The algorithm using 3 passes of overwriting.
- German Standard VSITR. The algorithm using 3 passes of overwriting.
- OPNAVINST 5239.1A. The algorithm using 3 passes of overwriting.

SafeIT Implementation Review

Based on a contractual agreement any potential client will be allowed to review the SafeIT source codes and implementations of algorithms. All clients interested in this option will be asked to sign a confidentiality agreement. No copies of the source code will be released outside the company.

During the review all possible tests and studies can be made. A client will be accompanied during three full days by two technicians from SafeIT Security.

Price for a source code evaluation is: USD 25.000.

Contact Information

Ardy Electronics Ltd
Telephone: +46 19 247010
Technical support: +46 19 247010
Fax: +46 19 247011

Corporate website: www.ardy.se
E-mail - information: info@ardy.se
E-mail - support: support@ardy.se

[i] http://fp.gladman.plus.com/cryptography_technology/rijndael/
[ii] http://www.schneier.com/blowfish-download.html
[iii] http://www.schneier.com/paper-twofish-paper.html
[iv] http://burtleburtle.net/bob/rand/isaac.html
[v] http://www.faqs.org/rfcs/rfc1321.html
[vi] http://www.faqs.org/rfcs/
[vii] http://www.faqs.org/rfcs/rfc2631.html
[viii] http://www.faqs.org/rfcs/rfc3174.html
[ix] http://burtleburtle.net/bob/rand/isaac.html
[x] http://fp.gladman.plus.com/cryptography_technology/rijndael/
[xi] http://www.schneier.com/blowfish-download.html
[xii] http://www.schneier.com/paper-twofish-paper.html
[xiii] http://burtleburtle.net/bob/rand/isaac.html