

White Paper

Encryption Method used in

DataCentric's

***GuardIT* Data Back-up service**

The encryption method used in DataCentric' GuardIT is known as BLOWFISH. This method was designed by Bruce Schneier who is a well known expert in the field of encryption. Mr. Schneier is also the author of the excellent book Applied Cryptography, Second Edition, (John Wiley and Sons, 1996) which gives the complete code and description of BLOWFISH as well as that for most of the other encryption algorithms in use today.

Technically BLOWFISH is a 16 pass block encryption method that works in blocks of eight bytes. During each pass the bits are shuffled in a pattern that is different for each locking key. After 16 passes the bits appear completely randomized. However, the seemingly random bytes can be recovered by rerunning the algorithm with the same encryption key. GuardIT performs the encryption of the source data when the "locked" safe is saved as a .SAF file. Each of the files listed within the safe are read in sequence and written in encrypted form to the .SAF file.

The encrypted .SAF files are ideal for electronic transmission because both the originator and recipients of encrypted data (the client's PC or server, and DataCentric's data center servers) are using the DataCentric program. If the recipients do not have the program GuardIT can be used to create self-extracting files. In this case a small Windows executable is written around the encrypted .SAF data. The recipient can unlock the data by just starting the self-extracting file within Windows and entering the combination. Security is identical with both native .SAF files and self-extracting files as the BLOWFISH algorithm is used in both cases. The only requirement for self-extracting files is that the recipients must be running some form of Microsoft Windows.

BLOWFISH is also used to protect each user's private list of stored combinations. The user's password is used as the encryption key. The saved keys are stored in the Windows registry in encrypted form.

Key Length and Security

Many people assume that the security of an encryption method is just related to the length of the encryption key ("locking combination" in DataCentric terms.)

The press has reinforced this misconception in their superficial coverage of issues such as encryption export limitations and the "clipper chip."

The reality of encryption security is that the issues are more complex. Let's take the issue of key length. GuardIT uses eight digits in the locking combination (sometimes called a "64 bit" key.) Each of the digits can have any of 256 possible values. This means that the number of possible combinations is:

$$256 * 256 * 256 * 256 * 256 * 256 * 256 * 256 @ \\ 18,446,744,070,000,000,000$$

This is a very big number. For example, if you linked together a group of very fast computers to all work on the problem and were able to test 1,000,000 combinations a second on a secure file, it would take almost 300,000 years to try all the combinations. If in that 300,000 years the user decided to change the encryption key you would have to start over again.

Obviously the 300,000 years could be made even longer by adding more digits to the combination, but the advantage is questionable. Faced with these impossible odds someone attempting to break into an encrypted file will tend to fall back on one of the following methods:

- Guessing obvious combinations.

DataCentric combats this to some extent by eliminating obvious combinations such as "password" and sequences of digits.

- Finding out the combination directly. Users have been known to share their passwords. (Some firms use security tokens to help combat this and the previous problem.)
- Statistical attack on the encrypted data.

Statistical attack is the real proof of an encryption method's security. A statistical method does not attempt to try every locking combination but instead attempts to determine the pattern of bit changes needed to reduce the randomness of the file. Poor encryption methods such as XOR masks can be decrypted rapidly with these methods. More sophisticated methods require more time. Strong encryption engines do such a good job randomizing the bits of the file that the statistical methods cannot find any pattern and the decryption team must fall back on guessing combinations (the brute force 300,000 year case again.) Although mathematicians can make intelligent guesses about the relative security of different encryption methods, the real proof is in the test of time. If many mathematicians working over many years cannot do better than trying every combination then the encryption algorithm is considered secure.

A number of excellent algorithms have been developed that have passed successive testing by many people. BLOWFISH was selected because:

- It has been repeatedly tested and found to be very secure.
- It is extremely fast due to its taking advantage of built-in instructions on the current microprocessors for the basic bit shuffling operations.
- It was placed in the public domain and can be distributed without limitation.