

# GED-I Ltd Technology White Paper

Jun 2009

## Today's Encryption Status

Encryption methods have developed and improved over the last decades, mainly for communication uses. In this field it is usually difficult to break encrypted information, mainly because of the relatively small volume of data, keys exchange and the very short period of time available.

These days, the ability to crack encrypted information has increased due to the techniques and tools that have been developed and used by determined parties and due to the advanced computing power now available.

The huge amount of data stored on storage devices is normally kept unencrypted. In cases in which it is encrypted, this is implemented with the standard, known methods (AES, 3DES) usually file oriented.

Due to the inherent characteristics of storage environment, there are inherent vulnerabilities that these solutions can't overcome.

Mere standard encryption is not strong enough to encrypt a huge amount of data in known data structure, known storage structure, of data permanently available on a storage device, data that can be recovered, data that can be intentionally written to the device, and the possibility to physically steal the actual device or copy the data and take it to another place where dedicated cracking tools can be use.

These storage environment characteristics enable to decrypt the data using strong computation platform while utilizing several methods based on statistical, heuristically and algebraic algorithms.

Some of the published methods and reasoning are:

- Linear Cryptanalysis – linguistic statistical analysis, known information on files, huge amount of data with known characteristics, short keys implementation, etc.
- Differential Cryptanalysis
- Side channel attacks: Timing attack, Collision attack, AES in counter mode attack, Power analysis, fault analysis.
- Mathematical: Hidden subgroup, sub-algebras, ...
- Computation power: Moore law, parallel-grid computation, Quantum computation

Therefore, as data storage is becoming increasingly larger, while computing power continues to boost up, and also due to privacy regulations, it is necessary to use an advanced and more powerful method that is specifically designed for storage media

### **Ged-i's Technology**

Ged-I Ltd unique technology for storage security comes to overcome the inherent vulnerabilities of the storage devices. It is based on three layers of data protection for "Data at rest" and "Data at Transit" for storage disks and tapes:

1. Standard Algebraic Encryption Method.
2. Segmentation and Scrambling.
3. Interference operation.

**The first Layer** utilizes a certified standard algebraic encryption method (one or more). Currently we utilize the AES standard method. This method is usually used for communication data encryption.

But, while implementing the AES method for storage application, we understand that this method should be implemented differently for storage than the way it is used for communication applications purpose. Implementing AES or other algebraic methods as it is usually implemented today, significantly reduces the encryption complexity power. The way we are implementing the AES to a storage device is part of our patent. In general, in our implementation, we use many independent algebraic keys, such amount which is needed to successfully encrypt huge amount of data. By doing so, we significantly reduce the efficiency of the statistical and heuristically methods of cryptanalysis, since the more independent keys are used, the statistical methods applicability is reduced.

Therefore, by having a lot of independent keys, we preserve the major logic behind the use of algebraic encryption method: short key per short communication session that involves small amount of data.

We are also compatible with the ongoing developed P1619 IEEE Standard for storage Encryption.

Today there is a common agreement that the used algebraic methods are safe enough when applied for small amount of data. That is keeping on the assumption that the encryption power is 2 to the power of the key length – i.e.  $2^{256}$  in the case of 256 bit key length.

The number of keys issued - is part of the company core technology. A unique formula was developed as to the number of independent algebraic keys one has to use. For example, based on our algorithm we use about 1.5M AES encryption keys for 1 Peta Byte .

During the history of the modern encryption domain we learned that algebraic cryptographic methods are usually weakened while time passes. This occurs as a result of the advances in: computation power (following Moore law), cryptanalysis (statistics, attacks, differential, noise & time analysis, etc.), Algebraic theory (e.g. by finding hidden sub-algebras), etc.

Therefore we find the need for additional layer to “encrypt” the data – The method is geometric hiding of the data.

**The second Layer** defines different geometry map uniquely for each storage device. The unique geometry map is generated randomly for each storage device and it is kept encrypted in remote device. The information about the hidden unique geometry of each storage device is never located on the storage itself. In this method we prevent any operating system to be able to fetch the information stored on the storage, unless the unique geometry map is available, accessible and decrypted.

In our method, from encryption point of view, the storage device is always full of encrypted data even if only part of it was actually used by the storage clients. There is no way to know whether specific data is fake (initiated by the system) or real. Moreover, because of the unique geometry per device, continuous data is fragmented. Therefore, prevent file reconstructing which is essential for using statistics in cryptoanalysis.

In more detail – Having a storage media, the system examine the storage media sizes and topology (i.e. the storage configuration). Based on this examination, the system automatically generates a random map for the storage.

There are two types of random map one can generate. The first type of random map is related to the physical location of the blocks (sectors) on the storage, and their sizes. In this case the randomness revealed in the block locations and/or in the block sizes. (See e.g. picture 1)

The second type of a random map generation is when an already map is given, and one randomly scrambled its block (sectors) addresses. While keeping the bijection (one-to-one) relation hidden. (See e.g. picture 2)

Of course one can apply the second type map on top of a first type map.

As to the induced complexity power of these types of geometric origin “scrambling”:

In the case of type 1 random map – the power is the power the continuum. That is mathematically symbolized as  $\aleph$ . It also can be expressed as  $2^N$  where  $N \rightarrow \infty$ .

In the case of type 2 random map – the complexity power is factorial in the number of the scrambled/shuffled block addresses. That is if one scrambles N addresses then the complexity power is N! (i.e. N factorial). Here we assume that all the blocks are exactly

similar to each other, and the data written upon them doesn't help in re-labeling the addresses. This is the case where the disk is fully filled and all the data is encrypted. In case that the algebraic encryption is broken or that all the data in each of the blocks is not encrypted, then the blocks can be distinguished, and by their content one can try to reconstruct files out of the blocks. In that case reconstruction power of data is then:

- Therefore, for the first file to be reconstructed, if the file is stored on  $k_1$  sectors, the chance of reaching and right ordering of them is:

$$COMB_1 = \frac{(N-1)!k_1!}{(N-k_1)!}$$

The decryption complexity of the  $i$ -th file, after all the former ( $i-1$ ) files have already been reconstructed is:

$$COMB_i = \frac{\left( N - \sum_{j=1}^{i-1} k_j - 1 \right)! k_i!}{\left( N - \sum_{j=1}^{i-1} k_j - k_i \right)!}$$

The total decryption complexity of the entire encrypted disk is therefore:

$$COMB = \prod_{i=1}^{N_f} COMB_i$$

Where  $N_f$  is the number of files within the encrypted disk.

We ignore in above calculation the fact that the number  $k_i$  of sectors of the  $i$ -th file consists of is also unknown. This also inserts a large factor to the complexity calculation.

**The third Layer** is idle time interference operation.

Data writing on storage devices is based on electromagnetic technology. A "new" data is characterized by Electromagnetic emission that is stronger than the "old" data emission. Accordingly it is possible to distinguish between "new" data and "old" data.

Today's electromagnetic analysis tools are so sensitive that they can be used for data dating, based on the strength of the magnetic field and other physical trace analysis.

The interference operation changes the Electro magnetic characteristic of the storage device. Thus, it prevents from sensitive electromagnetic tools to distinguish between "old" and "new" data and accordingly to defragment or reorder, the scrambled data as implemented in the geometrical restructuring, thus, reducing the complexity power of the random geometric map.

By randomly renewing the emission on the disk we keep the complexity power as expected according to the second level of the geometric restructuring.

The interference operation is executed by randomly renewing of data all over the disk. This process is even mislead any one who will try to use those instruments. This third level is executed only in system idle time.

Therefore the total complexity power of the encrypted disk will be the product of the complexities power of each of the levels.

