

Data storage that is as  
private as your thoughts



data**GRIT**

White Paper

Sergei Petrov

2008

## FOREWORD

We all have secrets.

There is no better place than our own mind to keep them safe, in the privacy of our thoughts. However, life in an age that is ruled by technology and information requires us to manage immense quantities of data. As our identities extend “on line” what we place in the cyberspace is becoming an important part of who we are.

What happens to our personal information, who has access to it, and how it is being used are key to guarding our privacy and fundamental liberties. Individuals and organizations, governments and businesses, we all need to be assured that our vital data is safe and secure, that we can share it with confidence, knowing that it can be accessed only by those to whom we have given our explicit permission.

Welcome to DataGrit: the first global data storage system that will guarantee the security of information and the privacy of its users. “Guarantee” is a strong word and I do not use it lightly. The risk of information theft and technological failure within DataGrit is so small that it could be disregarded by the users, along with other extremely rare events, such as being hit by a meteorite while reading this paper.

DataGrit data storage is truly as private as your thoughts. However, it is not a “silver bullet”, nor is it a “magic cure.” It derives its strength from converting information into “data grit”: indistinguishable blocks of data dispersed arbitrarily among countless storage units and sites. The technologies of DataGrit can guarantee security and privacy, but they do require extremely large amounts of data to be stored by the system to be effective. Just like the internet itself, DataGrit would rely on many independent service providers to maintain its gigantic pool of anonymous “digital pulp” spread among many storage facilities around the world.

To implement a truly global data service on such a scale will take time and substantial resources. The process of transferring DataGrit into the public domain must be well managed and organized so the new technology is not monopolized, fragmented, blocked or subverted. It will require a joint effort by many companies and organizations, from hardware manufactures to service providers. But most importantly it will require the support and determination of ordinary internet users, who must insist on change and refuse to accept that stolen identities or loss of vital information is the inevitable price we have to pay for the convenience of living in a digital age.

Today, the internet connects us all to such a degree that the security of cyberspace must extend to everybody in order to be truly effective. The same solution must work for everyone or it will be useful to none. DataGrit virtual data storage is of the internet, by the internet and for the internet.

Sergei Petrov

2008

## OVERVIEW

DataGrit can provide data storage to the entire multitude of current applications without interfering with their inner workings and interactions. The applications would “see” the DataGrit device as just another storage media, such as a disk drive or a memory key where information can be saved and retrieved. Any similarity with existing solutions, however, ends beyond the common interface.

When stored, the information is turned into “data grit”: anonymous, indistinguishable blocks of data dispersed arbitrarily among numerous storage units and sites. While this conversion is reversible and the information is returned to its original form when retrieved, the process is *arbitrary* in nature, where all blocks are overlaid with *true random* data and bounce from unit to unit as if in a gigantic pin ball machine. The result is one huge homogeneous data pool spread among many different locations all over the world.

The “data grit” approach would not be secure if implemented on a small scale as a stand-alone solution for an individual or a company. However, with enough data stored within the system it becomes unbreakable. A child can put together a few pieces of a broken plate, but trying to gain unauthorized access to the information stored in DataGrit is as hopeless as going through the grains of sand in a desert in order to reconstruct the mountains that stood there before.

Analogies aside, the risk assessment of DataGrit is straightforward math (combinatorics) that calculates the computing power required to go through all possible permutations of data blocks in storage.

The conversion of information into “data grit” is an iterative process that can be repeated as many times as necessary to achieve *any* desired level of security. This is a highly effective and efficient way of guarding access to data since with each step the storage costs increase only linearly, while the computing costs for a potential intruder grow exponentially. Thus the security of information and the privacy of users are assured by the very core of the system, without the need for extra layers of protection or additional technologies.

Designed with utmost simplicity, DataGrit is comprised entirely of basic hardware components. Since there is no software of any kind involved, it is immune to computer viruses or any other type of malicious programming. DataGrit does not require system administration and is free from perilous dependence on trusted human personnel.

The internal network of DataGrit is an amorphous cloud of functionally identical storage units that maintain its uniform pool of “digital pulp”. The use of the internet backbone is limited to direct point-to-point connections between the storage sites. The rules that govern data flow among DataGrit units have more in common with the *classical mechanics* of gasses and fluids than with conventional computer systems. Operational issues such as load distribution, redundancy or capacity growth resolve themselves in this kind of architecture with the ease of self-leveling liquid in interconnected vessels.

Everything in DataGrit, from conceptual design to implementation details, is so utterly simple and comprehensible that the system can be fully validated *analytically*, well ahead of the actual deployment.

## BACKGROUND

In its early years the security shortcomings of the internet were far outweighed by the novelty of creating an open, seemingly limitless digital domain. In time, as the internet became an essential means of communication, banking and commerce for our society, serious problems became more visible and annoying.

Today our situation can only be described as desperate: in spite of billions of dollars spent on cyber security each year the number of attacks and the damage they inflict continue to grow at an alarming rate. The threat is not limited just to hackers and criminals; companies are seeking our personal data to expand sales and marketing, while governments are trying to control the free flow of information.

There is no doubt that if the internet were designed from scratch with the benefit of contemporary knowledge and experience, it would be based on technologies that are far better suited to the job. After all, the intrinsic vulnerabilities of the original internet have been well known and documented for some time now.

Unfortunately, by the time the scope of the problem was recognized fully, it was already too late to go back and replace the technologies that were at fault; gradual modernization would make sections of the internet incompatible, while universal change is simply impossible, given the scope and the cost of the project. (We still drive on the opposite sides of the road in different countries and the rules for internet traffic are far more complex than for automobiles.)

As the world wide web continues to grow, numerous new services, products and applications provide cheap, effective means of reaching millions of customers, but their dependence on the underlying technology of the original internet makes them just as vulnerable.

Meanwhile cyber security has become a booming industry in itself, bringing billions of dollars in revenue and employing hundreds of thousands of people. However, it cannot give us the protection we need because of its own dependence on the very same technologies and architecture that are at the root of the problem. At best, we are provided with the equivalent of “painkillers”: they reduce the suffering somewhat, but do not offer a real cure.

The technological legacy of the original internet and its inherent vulnerability have proven to be extremely difficult to overcome. Making cyberspace secure remains one of the greatest challenges of our time.

## APPROACH

As the internet has matured, it is no longer defined by its original technologies, but by the role it plays in our civilization, by the way we use it. In the beginning, it was all about connecting computers, but now it is about connecting people: us. We, our privacy, and information are what require protection, and not a particular piece of software or hardware.

Since there is no realistic way of fixing numerous security holes in a multitude of specialized and localized data storage systems, DataGrit *bypasses* them altogether by providing the technological means for creating a *single* universal data service, available to all applications and all users: private, public and business alike.

Most important, this approach solves the legacy problem of the vulnerable technologies that are still in use by *complementing* them with the secure new service rather than trying to *replace* them within each and every application. This way anyone can switch to DataGrit at a time of their choosing and never lose compatibility with the rest of the internet.

Furthermore, as a truly global service, DataGrit would take full advantage of the gigantic size of modern cyberspace. By storing an enormous volume of information in a single distributed data pool, the system would pass the crucial threshold that is necessary for the practical implementation of its radically new technologies. These technologies can guarantee security and privacy of our information but they do require an extremely large amount of data to be stored by the system. Just like the internet itself, DataGrit would rely on numerous independent service providers to maintain this gigantic data pool, spread among many storage facilities around the world.

As much as we need to keep our information private, we also need to share it with others of our choosing. As the single service available to everybody, DataGrit would also act as a universal global data exchange where information can be shared among trusted parties without compromising its security or the privacy of the users.

## METHOD

The innate ability to hope for the best against all odds is indeed a great human quality. However, it is not a method that we can rely on when it comes to guarding the information that is vital to us. Though we do accept all kinds of risks in our daily lives, by choice or out of necessity, the security and privacy of cyberspace are not the areas where we want to take any chances. We require nothing less than absolute confidence that our data is safe and secure.

However our existence in a stochastic universe provides few absolutes. It is governed by probabilities, where just about anything is theoretically possible. Yet there is no reason to be concerned that your reading of this paragraph would be interrupted by a meteorite or that bouncing molecules in the surrounding air would suddenly all depart, leaving you in a vacuum. The odds of these phenomena occurring are so incredibly small that they play no role in our lives whatsoever.

DataGrit takes these kinds of extremely rare events as a natural benchmark for determining at what level the risk of information theft or privacy violation can be safely disregarded.

The actual method is threefold:

- First, in DataGrit all potential sources of *unpredictable, unforeseeable* risk (such as dependence on complex software or trusted human personnel) are eliminated by the utmost simplicity and transparency of its design and implementation.
- Second, the conversion of the information into “data grit” - anonymous, indistinguishable blocks of data dispersed arbitrarily among numerous storage units and sites - assures that the only remaining security risk is reduced to a single factor: the probability of finding matching pieces among very large sets of *true random* data.
- Third, DataGrit provides simple iterative procedures for reducing this remaining risk to *any desired level* by repeating the conversion process as many times as necessary. With each step the storage costs increase only linearly, while the computing costs for a potential intruder grow exponentially.

Thus security and privacy of information in DataGrit are guaranteed by statistical certainty and the improbability of large sequences of random events.

## PROCESS

Finding simple solutions for complex problems is difficult enough. But an even bigger challenge lies in retaining the clarity of original ideas throughout all phases of design and implementation. An initial straightforward approach can quickly degrade into complex monstrosity as more and more different technologies become involved when a new product makes its way to the market.

To achieve and maintain the utmost level of simplicity, DataGrit development followed a process that did not rely on additional technologies to resolve the practical issues of deployment. Instead, whenever presented with a new implementation issue, the original core ideas were revisited and revised to meet added requirements. The goal was to achieve more with less and to build the entire system, in all aspects and functionality, with a minimal set of prudently chosen design concepts.

Such meticulous work could only have proceeded as an open-ended search for the ultimate solution, without imposed deadlines and release dates. DataGrit took almost a decade to complete, but the result, in its simplicity and unity of design, is unprecedented in the history of the IT industry.

## CONCEPTS

The essence of DataGrit can be summed up in one short sentence: *turn information into anonymous digital pulp and scatter it at random among numerous storage units and sites*. This simple formula is empowered by five select design concepts:

- **Homogeneous Data Pool** comprises functionally identical storage units within an amorphous network cloud. Each unit contains linear storage media divided into standard size blocks and pre-filled with *true random* data. The head portion of a block serves as its permanent, system-wide identifier, the rest is data area.
- **Anonymous Routing** moves data blocks through the system by randomly chosen routes. However, the process is both *persistent* and *reversible*, so a given block can travel back and forth repeatedly between the origin and destination without the units having any knowledge of each other.
- **Random Data Flow** throughout the system is created and maintained by spontaneously moving data blocks among storage units.
- **Block Overlays** combine information to be stored with the blocks in *Random Data Flow*, effectively dissolving it in one huge homogeneous data pool.
- **Data Pressure** is the ratio between *empty* (random fill only) and *full* (mixed with user data) blocks that flow through a given point in the DataGrit network.

Together, these core concepts form the complete set of building blocks for all necessary functions and features of DataGrit. They permeate the design of the entire system from security and privacy to deployment and maintenance issues.

## DATA STRUCTURE

To be useful, information needs to be well organized. Modern data management tools rely on an intricate hierarchy of many different concepts and data formats. Some have evolved naturally from traditional paper archives, like files and folders, some have been imposed by storage technologies such as disk drives, while many others are application specific.

Nevertheless, no matter how complex or large, all these data structures can be represented by a set of standard size blocks that are linked together by a system of internal pointers. This is how the data is stored in the memory pages of a typical computer.

DataGrit extends this concept to a global data storage base spread over numerous storage units in many geographical locations. As far as applications are concerned, working with DataGrit is exactly like allocating and accessing virtual memory of practically unlimited size that is guaranteed to be safe, secure and available.

## DATA MOVEMENT

Each storage unit in DataGrit comes pre-filled with standard size blocks of true *random data*. The units spontaneously pick random blocks and send them to the other units, also chosen at random, thus creating a constant random flow of the blocks throughout the system. When a client device is connected to DataGrit it begins to receive a portion of the flow.

To store information, the device follows the following basic steps:

- Slice data item to be stored into chunks equal in size to the data area of the standard block.
- Overlay a slice with *true random* data from multiple blocks received from the DataGrit system (could be a simple logical XOR).
- Take one more received block and fill its data area with the overlay result.
- Send all used blocks back to DataGrit but retain their unique identifiers that serve as digital receipts to retrieve the slice of data.
- Repeat the process for all slices of a given item of information.

Although it seems to be quite different from traditional concepts of storage involving files, disk drives or servers, it is, in fact, a far more natural way for applications to store data. The block identifier acts as an address for a *virtual memory page* except that instead of being unique to only one computer, DataGrit block identifiers are unique throughout the system. This is guaranteed by the manufacturing process for the storage units when the identifiers are created.

To retrieve an item of information the process is reversed:

- Send the retained identifiers for a slice to the DataGrit system to receive the stored blocks.
- Restore the slice from retrieved blocks (could be just a repeat application of logical XOR).
- Repeat the process for all slices and then combine them into the original data item.

In DataGrit the initial flow of data is reversed. It is originated when the system sends “empty” blocks to all clients. As the clients store their data the “full” blocks return into the system and propagate a drop in *data pressure* from unit to unit. The storage units respond by sending out more “empty” blocks along low-pressure connections until storage requirements are satisfied and *data pressure* returns to normal.

This model of load distribution is simple, robust, and requires no centralized control or system administration. It is managed by individual storage units based entirely on the local state of their connections to immediate neighbors.

## DATA SECURITY

DataGrit's exceptional ability to reduce the risk of information theft or privacy violation to any desired level, no matter how low, comes directly from its extreme simplicity and the statistical certainty of its security model.

Imagine a well-organized warehouse where each numbered slot contains either a locked box or a key, distributed at random without any discernible pattern. If both of the two matching locations are known, getting the key and opening the corresponding box is easy. However, without such knowledge, a thief can only go through boxes and keys by trial and error. The more boxes in storage, the longer it would take to find a match, although the increase is only linear. Now, if a box has more than one lock, the multitude of possible combinations and the time to go through them grow *exponentially* with the number of keys required.

Naturally, the security of such a system would also depend on how difficult it is to break into a box without a key. In the physical world to protect such box would be quite a challenge, but in cyberspace the solution is simple: overlaying a block of information with random data of the same size. As long as "true" (no detectable patterns) and "unique" (used only once) random data is applied, it is not possible, even theoretically, to obtain original data (plaintext) from the overlay result (ciphertext) without knowing the exact random data that was applied.

The technique itself is not new: variations of it, usually known as "one time pads", have been in use for some time. What is revolutionary in DataGrit is that ciphertext and pad blocks are all mixed at random within a single homogeneous data pool, taking full advantage of the extremely large volume of data in its virtual global storage. The risk of unauthorized access in the absolutely worst-case scenario, when an entire storage pool is exposed, is determined by a single factor: the probability of finding matching pieces among very large sets of *true random* data.

A potential intruder is left with no other choice but a brute force attack against DataGrit. Since a linear increase in the number of random pads applied to a single data item makes the number of possible combinations grow exponentially, a reasonable increase in storage space will always be sufficient to compensate for advances in computing hardware available to an attacker and to maintain any desired level of protection against unauthorized access.

The very nature of DataGrit's anonymous data processing assures that neither a particular user nor a piece of information can be targeted selectively. By increasing the number of overlays the users can choose different levels of security according to their individual needs and available resources.

## DATA SAFETY

As the internet evolved, so did cyber crime. From simple blackmail to sophisticated stock manipulations there are ample scenarios where the objective is actually to destroy data rather than to steal it. An attack by a fundamentally new computer virus or a new cyber crime scheme, devised by a trusted employee, comes as a complete surprise and has no known history. The possible impact and associated risks of such events are, therefore, unknown and cannot be managed. To keep stored information safe, all sources of all unpredictable risk factors must be eliminated completely.

To be totally immune to any type of malicious programming DataGrit has been designed without the use of software of any kind. It is comprised entirely of basic hardware components that manipulate data flow according to the rules that are permanently “wired” into the system. The simplicity of DataGrit architecture requires the support of only a limited set of operations on data, making its storage unit much more closely related to specialized *signal processing* equipment than to traditional computers.

DataGrit also has no dependence whatsoever on trusted human personnel. The stochastic data flow is self-regulated by individual storage units that maintain uniform *data pressure* throughout the system. Operational issues such as load distribution, redundancy or capacity growth resolve themselves in such architecture with the ease of self-leveling liquid in the interconnected vessels and do not require system administration of any kind.

The remaining risks to data safety are associated strictly with hardware failures and can be accurately accessed given the homogeneous nature and large scale of the DataGrit system.

The risk is managed at two levels:

- First, all operations at every stage of the data flow are interlaced and performed by the redundant hardware components. The objective is to establish a clear base line for the risk of losing any specific data block in storage due to hardware failure within a given time period.
- Second, multiple copies of data blocks are being stored at randomly chosen storage units. Every unit has an independent sub-system that periodically verifies that all copies of the given block are still available. If not, a new copy is created so the total number remains at the original level.

The shorter the time period between the checks, the less the risk of losing data. In order to keep up with DataGrit’s highly efficient security model, the frequency of verifications is also set to grow exponentially with the number of copies created. Thus a user gets the same increase in data security and safety as the investment in storage space per given item of information.

## DATA ACCESS

Every storage unit in DataGrit also serves as a gateway to the internet, enabling the flow of data blocks in and out of the system through numerous access points. Any client of DataGrit can interface simultaneously with as many units as needed for the required rate of data transfer.

All data block transactions are atomic and independent, making it very easy to switch from one set of randomly chosen gateways to another, even in the middle of the transfer of a particular data item. Such architecture renders targeted denial of service attacks completely meaningless, while an attack against the entire DataGrit global storage service is simply impractical, given the huge scope of the system.

DataGrit's numerous access points are randomly scattered among many different service providers. As long as DataGrit users have an internet connection of any form, they are guaranteed to be able to store and retrieve their information.

## HARDWARE

The ability to run various software programs on the same hardware was the key factor in the computer revolution and the explosive growth of the internet. Such an approach is essential where flexibility is paramount and the software programs run millions lines of code to support all the required functionality.

On the other hand, DataGrit has been designed with such simplicity that the storage units need to perform only a few basic operations on data, which makes the strictly hardware approach the best by far.

The practical implementation of DataGrit storage units does not involve venturing into uncharted territory or working with unproven technologies. The typical unit is not a super computer but a mere “data brick”. All it has, besides the storage media, is a data bus, several independent network interfaces and a set of basic logic circuits that are responsible for moving the blocks in and out. All of these components have been in use for years and there is a vast knowledge base on their design and manufacturing. The creation process for the DataGrit storage units consists essentially of taking what already exists and stripping off all the extra functionality, thereby increasing reliability and reducing manufacturing costs for mass production. Removing all unnecessary features also assures that the system cannot be “turned into itself” in an attempt to gain access to the information in storage.

## COMPANY

The key to creating a system that can actually guarantee security and reliability is finding the simplest possible solutions to very complex problems. Large, convoluted systems cannot be made secure, because any meaningful risk analysis quickly becomes impossible as the complexity increases.

However, those of us who have tried, do know how extremely difficult it is to arrive at simple solutions. A committee cannot do it, nor can a corporate bureaucracy. What is required for successful creation of a radically new, “out of the box” solution is a small company, research team or just an individual with a vision and specific, clearly defined objectives. Obviously such a tiny commercial entity will never be capable of deploying the system on any significant scale and would have to re-invent itself by raising more capital, forming partnerships or utilizing any other methods of commercialization.

Nevertheless, regardless of how successful any of these business strategies might be, they all share a fundamental security flaw: by maintaining a monopoly on the technology the company itself becomes a “single point of failure.” Even giant, well-established corporations go out of business and the executives who run them do make bad decisions. The safety and privacy of our information are too important to be entrusted to just one company and a small group of individuals that may control it.

***Transferring DataGrit technology into the public domain is the essential step to assure that no intentional or unintentional action of any single company will undermine the safety and security of the data and privacy of the users.***

A project of such gigantic scale will require substantial resources and the participation of many different companies and organizations. Until firmly established, DataGrit technology will also need extensive legal protection from possible abuse, such as restrictive patenting or monopolization. It is also imperative to avoid fragmentation of the market and to create a single global service available to all internet users without discrimination.

The best way to manage the process is through the creation of a new specialized fund that would serve as a guardian, advocate and promoter of DataGrit.

## NEXT STEPS

Even a brief look at the daily news tells us that we are not only losing the war to protect our privacy and information online, we have not even won any battles of consequence lately. It is no longer a question of “if” or “when” but rather of “how soon” before a perfect storm of cyber crime may conquer cyberspace. We have grown so dependent on the internet in all aspects of our lives that the effect could be truly catastrophic: not only could it destroy economies, businesses, public institutions and ruin many lives, it could also tear the very fabric of our society and create social unrest on a global scale. The damage would probably be on the order of the global economic meltdown currently underway, with even wider implications that could defy hope of repairing it in any foreseeable future.

The lessons we are learning from the economic crisis tell us that we simply cannot wait any longer. We must act now.

DataGrit is seeking to assemble a small cadre of people dedicated to making cyberspace secure, who are strategically placed in our society to turn the urgent need for security and privacy in cyberspace into a coherent movement behind a real, practical solution.

The immediate objectives are to:

- Generate resources to initiate the transition of DataGrit into the public domain.
- Introduce DataGrit technology to hardware manufacturers and service providers.
- Promote DataGrit with the internet community and make the users aware that the solution to information security and privacy is possible, available and forthcoming.

The utmost simplicity of DataGrit and the very nature of its technology assure that it can be validated, manufactured and deployed very quickly with minimal disruption to existing commercial internet relationships. Its major challenge will be in mobilizing public support for a project of global proportions. The ultimate success thus will depend on the help and participation of us all.

## CONTACT

DataGrit

Sergei Petrov, Founder

[contact@datagrit.com](mailto:contact@datagrit.com)

[www.datagrit.com](http://www.datagrit.com)