Galleon Whitepaper

Secure Erase vs Encryption

Revision history:

Revision	Date	Changes	Author
0.1 draft	28-MAR-14	Document created	EB
0.2 draft	08-SEP-15	Updated information on SEDs.	EB
0.3 draft	06-JUL-16	General updates	HT
1.0	07-JUL-16	First release	HT
1.1	25-SEP-18	More details for encryption options	HT
1.2	12-OCT-18	Fix document number	HT

Abstract

This document describes the methods of Secure Erase and Encryption for securing and/or permanently deleting data in sensitive applications. The most important benefits and concerns with the two methods are discussed.

1 Introduction

Securing data at rest is an important aspect of many modern data systems in military, aerospace and commercial use. This whitepaper will take a closer look at the various options for securing and permanently destroying data at rest.

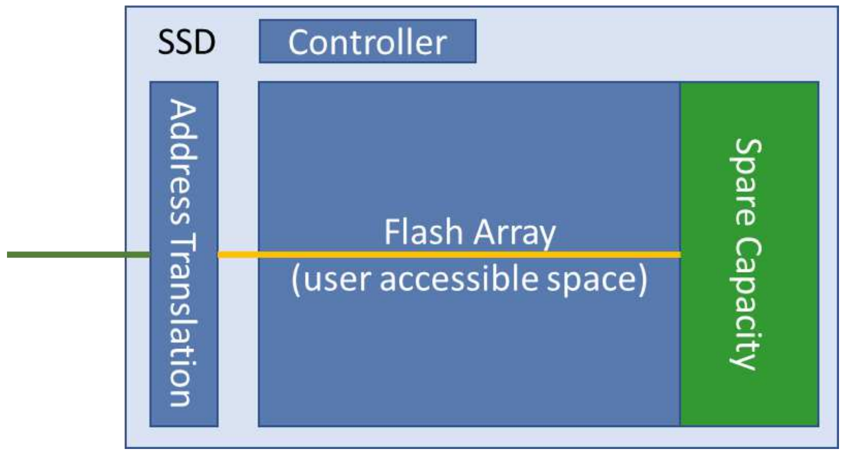
Galleon offers two methods for data security: Secure Erase functionality and Data Encryption. In some applications, both methods are used together.

2 Secure Erase

Secure Erase is a method for permanently erasing data on a storage media in a way which makes it impossible (or at least extremely difficult) for anyone to restore once the secure erase procedure has completed. The secure erase function is implemented in the Solid State Storage (SSD) media, and consequently only available on models supporting this functionality. These SSDs include high-end military grade SLC and some industrial grade MLC devices.

Note that traditional software based secure erase algorithms will not guarantee total sanitation of solid state storage media due to the over-provisioning used on these drives. In short, to mitigate issues related to memory cell wear-out, the SSDs have more FLASH memory than the capacity stated on the drive. Typically, 5 to 15% of FLASH memory is reserved for this purpose. The FLASH controller will use this additional FLASH memory to replace cells starting to show signs of wear. Since these memory cells are not accessible to the user application, secure erase SW algorithms will not be able to erase these blocks, and sensitive data may be left behind in unused cells on “erased” drives.

Figure 1, SSD spare capacity



To address this issue, the Secure Erase algorithms must be implemented in the SSD controller firmware. The available algorithms vary with the drive model chosen. Secure erase functionality may include certified algorithms, such as DOD 5220.22, NSA/CSS Manual 9-12 and RCC-TG IRIG 106-7 Chapter 10 as well as fast erasure techniques.

2.1 Erasing Contents vs Allocation table

Note that the term “fast erase” is sometimes used to describe a process where only the allocation table of the FLASH controller is erased, and not the contents of the memory cells. In this case, the actual contents of the FLASH memory cells is left intact and could be read if the memory is removed from the device and accessed directly. This is opposed to a “secure erase” algorithm which physically clears each memory cell to an erased or other known state. The term “military erase” is often used as reference to more advanced algorithms where each memory cell goes through a number of erase and overwrite cycles to ensure no user data is left behind in a readable state. Throughout the rest of this document, the terms fast erase, quick erase, and Zeroise refer to a fast versions of secure erase – i.e. erase of the complete disk contents.

2.2 Secure erase algorithm options

Flash memory is designed to operate on block basis, where all memory cells in a block are erased together, and individual cells may be written individually. It is not possible to erase a single cell. This restriction has some impact on the longevity of the device (refer to GEC-WP-1403, Galleon’s white paper on SSD technologies and flash memory wear out).

However, the block erase also helps with very fast complete disk erase. Running SSD erase based solely on flash memory block erase commands (issued by the SSD controller internally) is called ‘zeroise’, ‘fast erase’, or ‘quick erase’ (in accordance with notes in section 2.1 above).

Different SSDs implement different options for the other available algorithms. Note that many of the available options for erase algorithms were developed for magnetic media (HDDs), and it is unclear whether they actually offer much improvement on security of the data compared to simple flash erase (e.g. http://cseweb.ucsd.edu/~swanson/papers/Fast2011SecErase.pdf). In all cases, the SSD controller issues commands or writes data to the flash memory array, without any further interaction with the SSD host.

2.3 Initialising erase and erase duration

Galleon offers multiple methods for initializing the erase function, including:

- Discrete input
- API command
- Combination of discrete input and API command

Inside the Galleon product (XSR, G1, offload server or docking station), these commands are used to issue the correct command to the SSDs themselves.

Unfortunately, there is no standard defined for how to initiate the secure erase function. Consequently, the activation and algorithm depends on the vendor implementation. Some offer a way to trigger the secure erase through a discrete signal which is made available in a front panel connector on the SSD, or through one of the maintenance interface connections present on the device. Some also support activation through a custom SATA command issued by the host OS.

The time it takes to perform a secure erase highly depends on the selected SSD manufacturer and type, as well as the algorithm required. The below table lists a number of algorithms and the related erase times for a representative drive.

*The times listed in the table below are given as estimates which apply to both hardware and software triggering of the erase sequences (where applicable). The actual time to complete varies with FW implementation, SSD vendor, FLASH type, software overheads, etc.

Table 1, Erase performance Table.

Erase Methods	**Erase Time as a function of SSD Capacity**			
	128GB	256GB	1TB*	2TB*
D0h Clear	3s	3s		
Quick/Fast Erase	3s	3s	10s	15s
DOD 5220.22	8m	15m		
DOD 5220.22 Sup1	25m	48m		
NSA 130-2	25m	46m		
ARMY AR380-19	25m	47m		
Navy NAVSO P-2539-26	17m	31m		
Air Force AFSSI-5020	150m	266m		
NSA 9-12	9m	15m		
IRIG 106-07	35m	56m		

Note: For 1TB and 2TB, only quick erase numbers are available at time of release.
Also note that these times are for individual SSDs, not for a four disk RDM.

2.3.1 Erase completion

For all algorithms and SSD based solutions offered by Galleon, once the erase command is issued to the SSD, the SSD controller will ignore all other commands until the erase process is completed. This applies even if power is removed. Once power is restored to the SSD, the controller will continue the erase process to completion.

2.4 Physical erase/destruction options

Some SSDs are available which implement a physical destruction methodology for data security. The mechanism employed is to apply very high voltage and high current to the flash memory cells. Theoretically, this will ensure that it is impossible to recover any data from the device.

Galleon Embedded Computing does not currently offer any of these devices within our recorders/NAS/servers, for 2 main reasons:

- The power draw required for the flash memory destruction is very high, and would require a much larger power supply and power rating for the product. This would require a larger enclosure, to the detriment of all other specification items.
- More importantly, the manufacturers of these physical destruction SSDs have not been able to justify any reliable form of verification. Verification of the physical destruction method has (for the most part) been by confirming that the SSD is no longer accessible, or that the flash memory devices are no longer accessible. Neither of these techniques actually verifies that the flash storage cells have been destroyed (or that the same result would occur with all devices without exception). In the absence of clear evidence of verification of the destruction methods being used, Galleon cannot recommend these devices.

3 Encryption

An alternative (or in some cases addition) to secure erase, is encryption. Depending on the application, the encryption option may represent a better solution for securing the data at rest.

Encryption offers enhanced functionality over secure erase in that the data is protected but not deleted from the SSD. E.g. when transferring the RDM (Removable Data Module) from an XSR recorder/NAS to a base station, the data is encrypted.

If power is lost or removed, the data cannot be retrieved until the key is loaded again. Since the key only resides in volatile memory, deleting the key is performed in a few microseconds, so securing the data is an instant operation if a threat is present.

Should the key be destroyed, the data is still intact on the SSD, and can be retrieved by re-loading the key. Hence, in the event a threat is present and the data is protected by erasing the key, the data is not permanently lost and can be retrieved at a later stage when the threat is no longer present. In the event a secure erase has been initialized, the data (and the key) will be permanently destroyed and is not recoverable.

There are two types of encryption commonly used: Self-Encrypting Drives (SED) and system level encryption. These two types are further discussed in the following sections.

From a high level, self-encrypted drives provide by far the simplest method for encryption, and require very little system level integration or control. However, system level encryption provides a higher level of security where it is required.

Both methods use standard encryption algorithms (AES-256).

3.1 Self-Encrypting Drives

Some SSD vendors offer “self-encrypting drives” (SED). In many cases, when there are no strict requirements for a high level of security and advanced key management, SEDs represent a simple and cost effective solution.

Self-Encrypting Drives have built in encryption circuits which are an integral part of the drive hardware and totally transparent to the end user when enabled. The methods and terminology used varies with the SSD vendor, but the fundamental process involves locking and unlocking the SSD.

Self-Encrypted SSDs apply encryption/decryption as part of their standard operation (typically with very little impact on performance). Such encryption and decryption is unnoticeable to the user, as such. That’s because all data which is written to the SSD is encrypted, and all data which is read from the SSD is decrypted. The encryption key is auto-generated by the SSD.

The security functionality of the SEDs is enabled by locking and unlocking the device. Locking involves writing a special code (key/password/passcode) to the SSD using a special command. The device will then be inaccessible until it is unlocked (by writing the same code to the SSD, with a different special command.

Note - typically, this key/password/passcode is not stored inside the SSD. Instead, it is used to encrypt/decrypt the encryption key in the SSD (randomly created inside the SSD). That encryption key is then used to encrypt/decrypt the data. This double layer of encryption affords some additional data security.

In normal circumstances, a disadvantage of SEDs is the need for local interaction at boot time, where the user is required to enter a passcode during the pre-boot initialization of the SSD. However, Galleon's implementation provides other options:

Table 2, Self-Encrypted SSDs unlock options

Passcode transfer method	Advantages	Disadvantages
Stored locally	Simplest method – removes system level complexity RDM security is assured (when separated from XSR)	The system (XSR and RDM) is not secure if accessed together.
Sent over secure link	Enhanced security (similar to system level encryption option – see below)	System level complexity Passcode could be intercepted
Manual entry	Enhanced security (but dependent on human security levels)	Generally not possible in embedded systems (requires keyboard or other HID) People sometimes write passcodes down, which reduces security

Galleon provides utilities and (in some cases) custom BIOS implementations on our hardware to ensure that these different methods are possible. However, each implementation is typically unique (depending on the system integration requirements). Contact Galleon sales representatives to discuss your requirements.

Passcode exchange depends on BIOS support or OS interaction. Further, the implementation will be SSD specific and changing SSD vendor or SSD type at a later stage may require modifications to the system (e.g. utilities and/or BIOS updates).

3.1.1 Secure erase with encryption

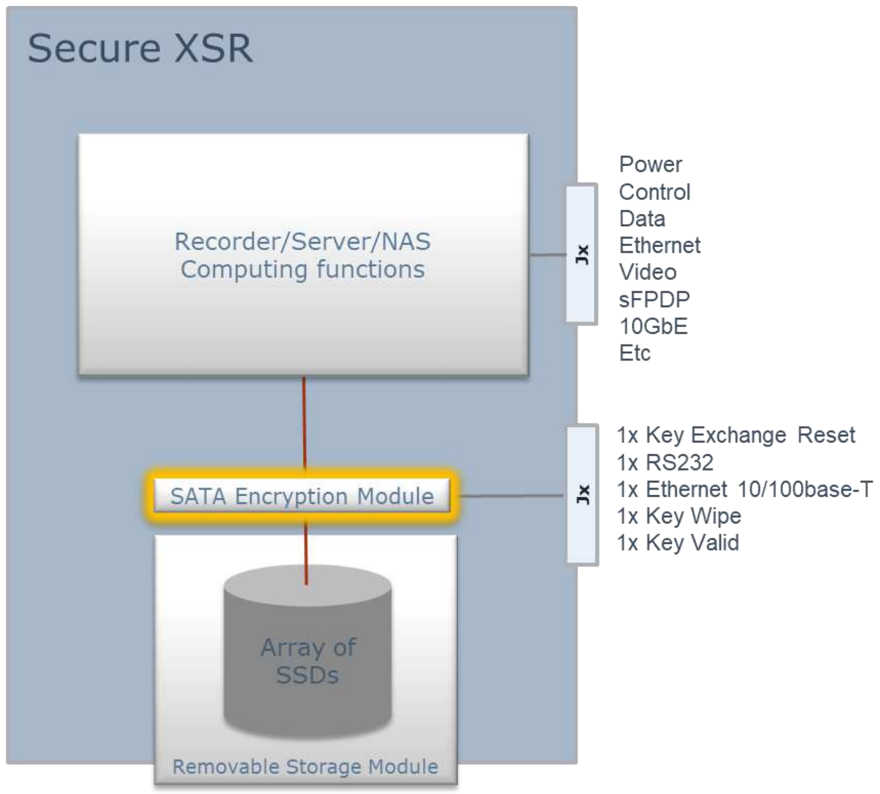
Galleon also offers RDMs based on SSDs which implement both secure erase and encryption. When erase is applied, the key (and any reference to the passcode) are destroyed as well as the data in the flash memory array. When the erase process is complete, the SSD generates a new encryption key ready for locking/unlocking to be initiated with a new passcode.

3.2 System Level Encryption

Where a combination of Self-Encrypted SSDs and secure erase do not afford sufficient data protection, Galleon offers advanced data encryption (AES 256-bit) implemented by an inline hardware encryption device. The encryption key is loaded from a key token, over RS232 from a local key management computer, or over a secure Ethernet connection.

Encryption/decryption takes place in dedicated hardware on the Galleon XSR secure server or data recorder. The encryption module is a self-contained unit which handles key exchange fully transparently to the operating system and the end user, as shown below.

Figure 2, SSD spare capacity



When the removable data cartridge is extracted from the server, it is physically separated from the encryption hardware, making it safer to transport to the final destination where it is re-installed in a server or docking station with the same encryption HW installed for data retrieval.

Finally, due to the encryption hardware being a part of the XSR server or data recorder, it is independent of the SSD chosen and will work on any SSD installed in the system. This allows for future upgrades to the latest SSD technology and capacity with no requirements to stay compatible with the encryption functionality embedded in the SSD. Further, tailored storage media can be used depending on the application. For example low cost commercial grade SSDs for lab and development use and high end military grade SSDs for deployed missions will all be supported by the same encryption hardware.

3.2.1 Comparison with Self-Encrypted SSDs

System level encryption provides additional data security compared to self-encrypted SSDs because the encryption key is not stored inside the disk and the encryption engine is inside the XSR (not inside the SSDs). So although the RDM has encrypted data stored on it, neither the key nor the encryption engine are present on the RDM.

The encryption key exchange is very similar to the case of self-encrypted disks when the passcode is sent over secure link, although even then, the encryption module has an advantage, because the Ethernet link used is dedicated to the microcontroller which is dealing with the encryption, rather than the general processor (which will also deal with the recording function, etc.).

4 References

Web article "**Commonly Asked Questions and Answers on Self Encrypting Drives**", (Trusted Computing Group), https://trustedcomputinggroup.org/resource/commonly-asked-questions-and-answers-on-self-encrypting-drives/

Web article "**Self Encrypting Drives**", (by Warwick Ashford, ComputerWeekly.com), http://www.computerweekly.com/feature/Self-encrypting-drives-SED-the-best-kept-secret-in-hard-drive-encryption-security

Web article "**ATA Secure Erase**" (Linux ATA Wiki), https://ata.wiki.kernel.org/index.php/ATA_Secure_Erase

Web accessible paper: Reliably Erasing Data From Flash-Based Solid State Drives
http://cseweb.ucsd.edu/~swanson/papers/Fast2011SecErase.pdf