



“ For over a decade Coolspirit have been supplying the UK’s top organisations with storage products and solutions so be assured we will meet your requirements head on.

It’s all about getting things right first time, quickly and simply! ”

Damon Robertson
Coolspirit Ltd

Our address

24 The Bridge Business Centre
Beresford Way
Chesterfield
S41 9FG

Get in touch

Call us on: 01246 454222
Email us: web@coolspirit.co.uk
Find us: [View location map](#)
Web: www.coolspirit.co.uk

Office hours

mon - thurs 8:30am - 5:30pm
fri 8:30am - 5pm
sat - sun Closed

“ Boost your storage buying power...
use ours! ”

Buy with confidence from
Coolspirit your authorised
HP Partner



Encryption technology for HP StorageWorks LTO Ultrium Tape Drives (LTO-4 and LTO-5)

Technical white paper

Table of contents

Introduction.....	2
Encryption basics.....	2
Using encryption in data protection.....	2
Encrypting data-at-rest.....	3
Overview of how encryption works in the HP LTO Ultrium Tape Drive.....	4
Cryptographic algorithms.....	5
Message digest.....	5
Secret key or symmetric encryption.....	7
Public key or asymmetric encryption.....	8
Encryption standards.....	9
HP StorageWorks LTO Ultrium Tape Drive encryption.....	10
Additional security features.....	12
Practical use of HP LTO Ultrium Tape Drives with encryption.....	14
Notes on key management.....	16
Conclusion.....	17
Appendix A. AES encryption.....	17
Appendix B. SPOUT parameters.....	18
Appendix C. Modular arithmetic.....	19
Glossary.....	20
References.....	21
For more information.....	21



Introduction

This white paper introduces the concept of encryption technology and its application in the HP StorageWorks LTO-4 and LTO-5 Ultrium Tape Drives. It also describes the basic cryptographic functions which are used in a tape encryption solution.

Encryption basics

Encryption is derived from the Greek word *kryptós*, which means hidden. It is the process of concealing information from unauthorized parties by means of a mathematical cipher, an algorithm that disguises the underlying text unless the reader has the de-cipher code. In encryption technology, the cipher is a complex mathematical algorithm that is applied to the unencrypted data, also known as “plain text”, to produce encrypted data known as “cipher text”.

A simple example of using a cipher would be to apply a straightforward alphabetical substitution such as replacing each letter of the alphabet with a different letter, for example, A=D, B=I, C=J, and D=Z. Unfortunately this method is very limited as a quick analysis of the letter frequency would easily reveal the substitution code used. More complex ciphers change the substitution each time it is used, for example, AAAA is encrypted as DFGT or similar random set of characters instead of, for example, HHHH the simplified A=H substitution. The above class of cipher is known as poly-alphabetical. Clearly far more complex mathematical algorithm ciphers are used today to ensure that the code cannot be broken by force.

To decrypt a message, a key is required as well as the correct mathematical algorithm. When the key is changed, the cipher completely alters the substitution sequence used. The correct key is required for recovering the original plain text.

The concept of encryption, in some form, is at least two thousand years old and was widely used in Roman military communications. Over the centuries mechanical and electromechanical encryption devices have been used in both commercial and military applications. Today, modern electronics and computers allow encryption (and decryption) to be performed in software and hardware with much higher speeds and using far more complex ciphers. More recent technology breakthroughs have enabled higher levels of security than ever before and resolved some of the concerns over encryption such as reliable key distribution.

Using encryption in data protection

Data security regulation aimed at protecting the disclosure or loss of personal data is now in force on an international landscape. These regulations increase the requirement for businesses to perform due diligence and care with the data in their possession with some regulations incorporating penalties for businesses failing to comply. Some examples of legislation concerned with data protection include:

- Sarbanes-Oxley (US)
- Gramm-Leach-Bliley Act (US)
- USA Patriot Act
- European Union Data Protection Act
- AIPA (Italy)
- GDPdU and GoBS (Germany)
- Electronic Ledger Storage Law (Japan)

The real cost to a business reaches beyond fines and penalties. Data loss costs millions in lost revenue, loss of customers, intellectual property, and damage to the brand. Most online data is protected by restricting access to applications by way of user administration, or by physical access to the servers and online storage located in a data center. The internet and intranet are carefully separated by hardware and software firewall technology. However, it is a common and necessary practice to remove backup tapes from the data center and into secure offsite locations for disaster recovery purposes. These storage sites may be run by third-party companies, and tapes in transit can be lost or stolen.

On February 9th 2007 a Baltimore Maryland University Hospital in the U.S. reported that computer tapes containing personal data on 135,000 employees and patients were lost in transit. Unfortunately news reports such as this have continued with over 95 reported data breaches in the U.S. alone during 2008, causing a staggering 285 million personal records to be compromised. A study by Verizon Business, published in 2009, found that over 60% of data loss instances were due to human error. As a direct consequence of these data breaches, over 34 U.S. states have introduced laws which force public disclosure by companies if tape loss occurs when the data is on the tape is not encrypted.

The threat to the security of personal information caused by lost or stolen tapes intended for archive is very real with each incident potentially costing the business or organization responsible millions of dollars to repair the damage caused. Encrypting data backups means data is equally secure when stored offline away from the data center.

“Have we not learned from history yet? If you’re going to give (data) to a third party then you either encrypt or password-protect it?”

Linda Foley, Executive Director of the Identity Theft Resource Center, San Diego

Encrypting data-at-rest

Data-at-rest can be secured using encryption by three different approaches; on the server using software, in an appliance between the server and the drive, or by the tape drive directly onto the storage media.

1. Backup software based encryption

A wide range of encryption software exists with a number of respected backup applications also featuring encryption capabilities. Unfortunately this method of encryption uses significant host processing power creating a performance hit both on the speed of backup and on the processing power available to other applications running on the server while encryption is performed. An additional complication arises from the fact that compression has to occur prior to encryption with this two-step process using even more processing power.

2. Encryption appliances

There are several dedicated encryption appliances available which sit “in-line with” the data flow between the server and the drive to offload the encryption from the host processor and so reduce the impact on performance. While performance is clearly protected, the cost involved with adding these appliances can be substantial at around US \$20,000 each. When you consider that several appliances may be required to serve a multi-drive library the management and scalability issues become apparent.

3. HP StorageWorks LTO-4 and LTO-5 Ultrium Tape Drive encryption

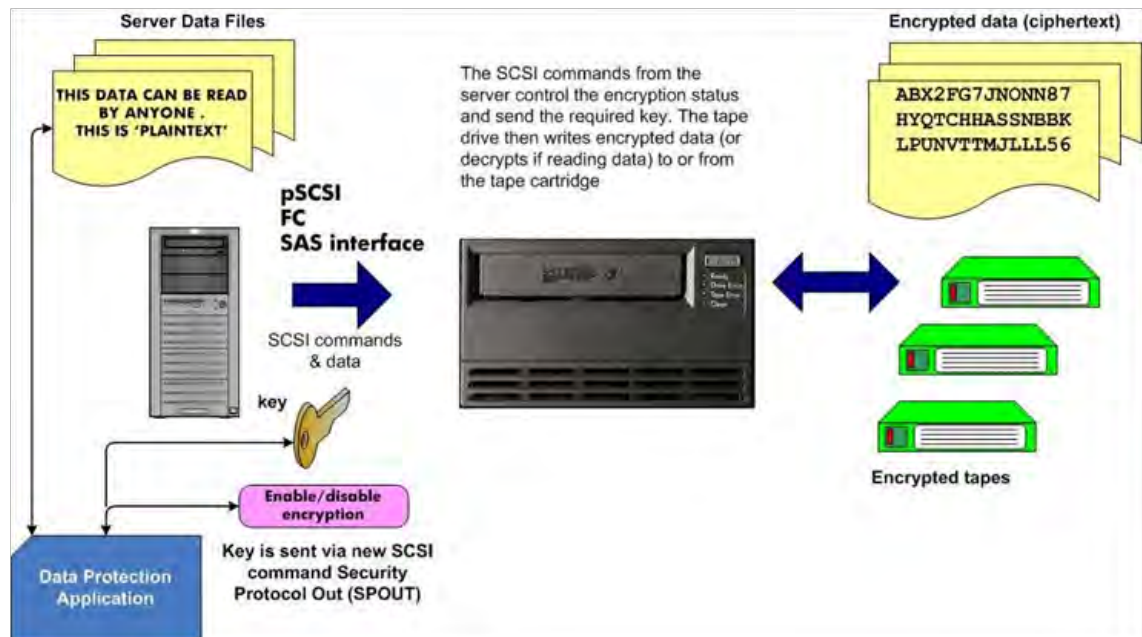
The open standard LTO format specification for both LTO-4 and LTO-5 includes the ability for data to be encrypted by the tape drive hardware. This adds a strong measure of security to the data stored on tape media without the process overhead or performance degradation associated with host-based encryption, or the expense and complexity involved with a dedicated encryption appliance.

Overview of how encryption works in the HP LTO Ultrium Tape Drive

The HP StorageWorks LTO Ultrium Tape Drive encryption is specified as part of the LTO-4 and LTO-5 open standard format with the Advanced Encryption Standard -Galois Counter Mode (AES-GCM) algorithm implemented in the tape drive formatter electronics. The implementation supports the IEEE P1619.1 standard for tape-based encryption and the T10 SCSI command set.

The diagram in Figure A provides an overview of the HP LTO Ultrium Tape Drive encryption process. Data files are taken from the server and pass through the SCSI interface to the tape drive. The SCSI commands issued by the server control the encryption status and the cipher key. The tape drive then encrypts and compresses the data before writing it to tape as ciphertext.

Figure A: Overview of the HP StorageWorks LTO Ultrium Tape Drive Encryption process



The remainder of this white paper discusses the technical detail behind the LTO-4 and LTO-5 encryption process and provides a primer on basic cryptography.

Cryptographic algorithms

All cryptosystems are based on three cryptographic algorithm techniques:

- Message Digest also known as a hash technique, Message Digest simply maps plain text of any length into cipher text of a fixed length. This technique is widely used for primitive security checks for message integrity, digital signatures, or for password verification. The flaws in this system include the fact that there is no usage key, and it is impossible to recover the original plain text. Typical implementations include SHA1 and MD5.
- Secret Key or symmetric encryption is the technique employed as the LTO-4 and LTO-5 encryption method; it uses one key for encryption and decryption. Symmetric encryption is subdivided into two classes, block ciphers and stream ciphers. Stream ciphers encrypt character by character providing a continuous stream of encrypted data whereas block ciphers operate on discreet blocks of data. Secret key/symmetric encryption is best suited for large amounts of fast moving data, usually encrypted in blocks, supporting the needs of high-performance applications. Blowfish, Defense Encryption Standard (DES), triple DES, and Advanced Encryption Standard (AES) are typical examples of secret key encryption algorithms.
- Public key or asymmetric encryption as the name suggests, this technique employs a pair of keys, one private and one public. The private key is kept secret, while the public key may be widely distributed. The keys are related mathematically, but the private key cannot be practically derived from the public key. A message encrypted with the public key can be decrypted only with the corresponding private key.

To fully understand the components of a tape-based data protection solution with encryption, knowledge of these algorithms is necessary. In the following sections, each type of cryptographic algorithm is examined in more detail.

Message digest

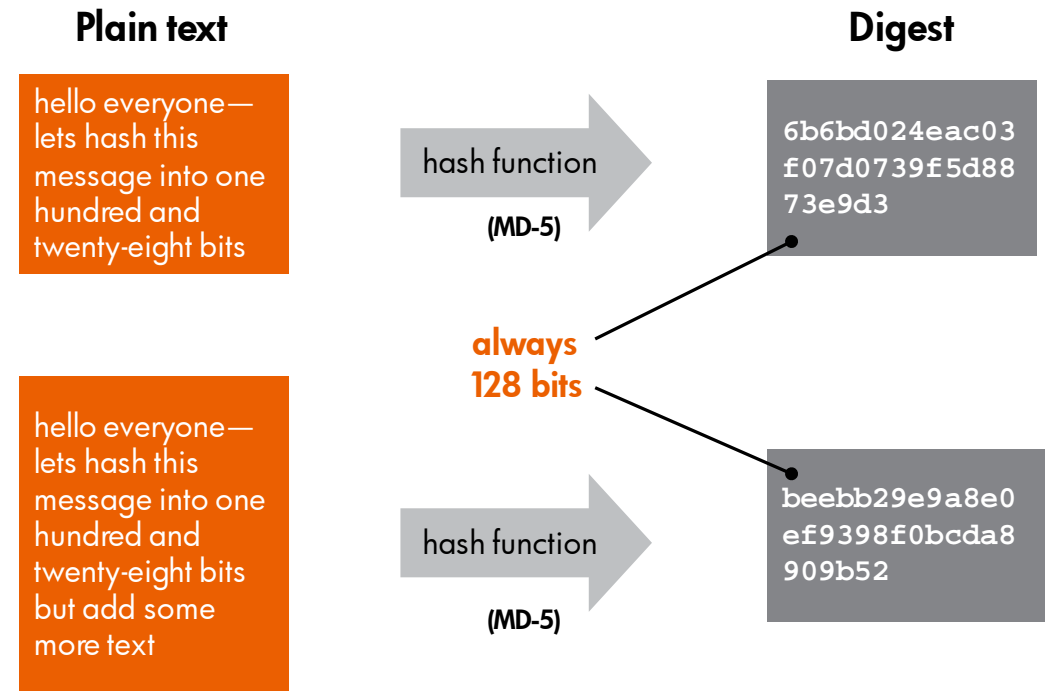
Message digests or hash functions, also known as one-way hashes, take a large string of data and convert it mathematically into a fixed-length string. Hash functions are known as one-way functions because you cannot recreate the original data from the hash. However, they are unique and provide a form of digital fingerprint for a message or data string. This technique is also used for checking passwords. The hashes of the passwords are kept in a file, and when a username and password is entered, the result is compared with the version on file. The basic requirements of a hash function $H(x)$ are:

- The input is any length
- The output has a fixed length
- $H(x)$ is relatively easy to compute from x
- $H(x)$ is one way (it is impossible to compute x given $H(x)$)
- $H(x)$ is collision free (the same hash is never repeated for different x)

Cryptographic hash functions are used to prove message integrity and create digital certificates. We will leave the complex mathematics behind hashing algorithms to another day, but if you wish to know more these algorithms are well documented and widely available through search-engines.

There are several types of hashing algorithm including the extensively used standards of SHA1 and MD5. Many UNIX® operating systems include a hashing function, and the hashes in the following example were produced on a SUSE Linux system using the basic operating system command, md5sum testfile. Figure 1 shows that even though the text files on the left are different sizes both digests are 128 bits long and despite similarities in the text content both digests are very different. The plain text cannot be recovered from the hash value. However, if the plain text is altered in anyway (even by one bit!), a new different hash value will be generated. Later in this white paper, the uses of hashes to digitally sign data are discussed with their use in Public Key Infrastructure (PKI).

Figure 1: A simple hash

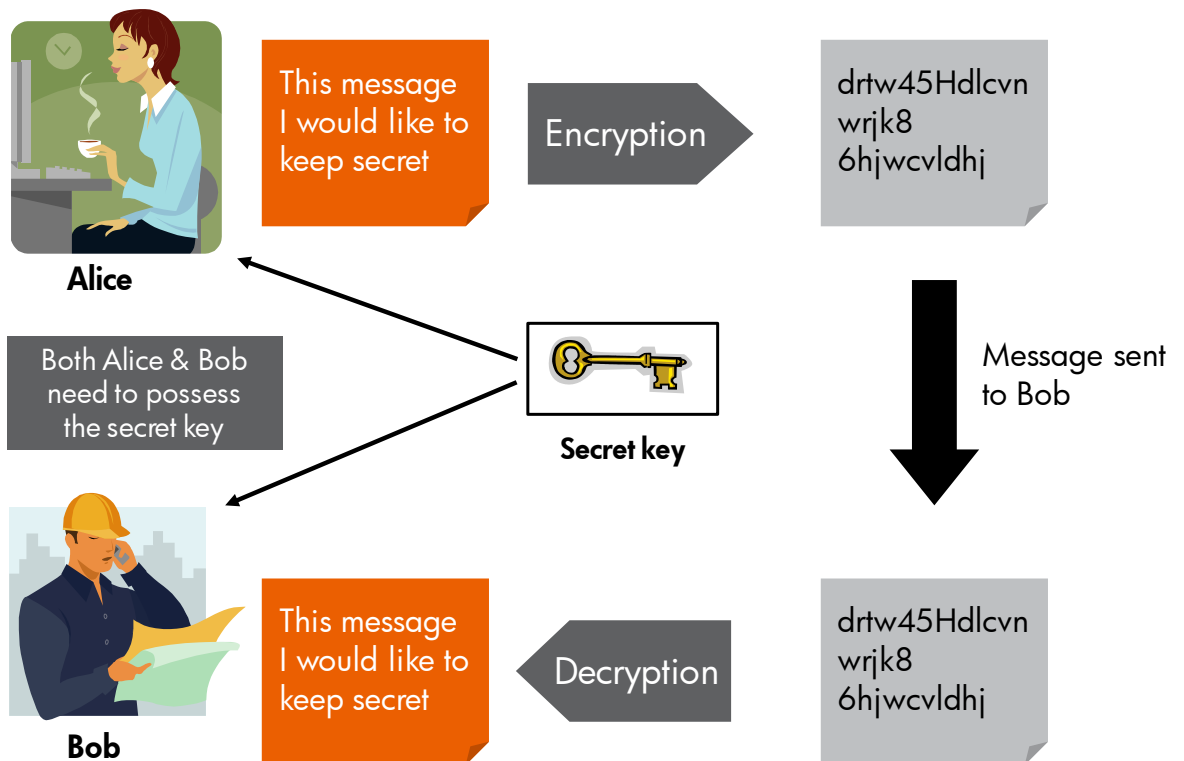


Secret key or symmetric encryption

This technique uses one key with an encryption algorithm to both encrypt and decrypt the plain text. In Figure 2, Alice wants to send a message to Bob, but does not want it read by third parties. Alice encrypts the message with the secret key and ensures Bob has the same key to decrypt the data on receipt.

Clearly this raises a number of security issues in terms of keeping the keys secure (we will discuss key generation and security later on in this white paper). This issue is offset by the benefit of secret key/symmetric encryption in accommodating large amounts of fast moving data by operating at a block level. This feature enables the high-performance encryption required by LTO-4 and LTO-5 applications.

Figure 2: Secret key or symmetric encryption



The algorithms used in symmetric encryption are two-way, meaning that decryption is the reverse process of encryption. Symmetric block-level encryption, referred to sometimes as a block cipher, is always used when large continuous streams of data need to be encrypted. The data to be encrypted is divided into blocks or groups of characters and the mathematical functions applied to each block. There are many block cipher designs such as Blowfish, the Data Encryption Standard (DES), Triple DES, and the Advanced Encryption Standard (AES) which has now superseded DES. The key length varies according to the type of cipher with DES having 56-bit keys and AES having 128-, 192-, or 256-bit keys.

Public key or asymmetric encryption

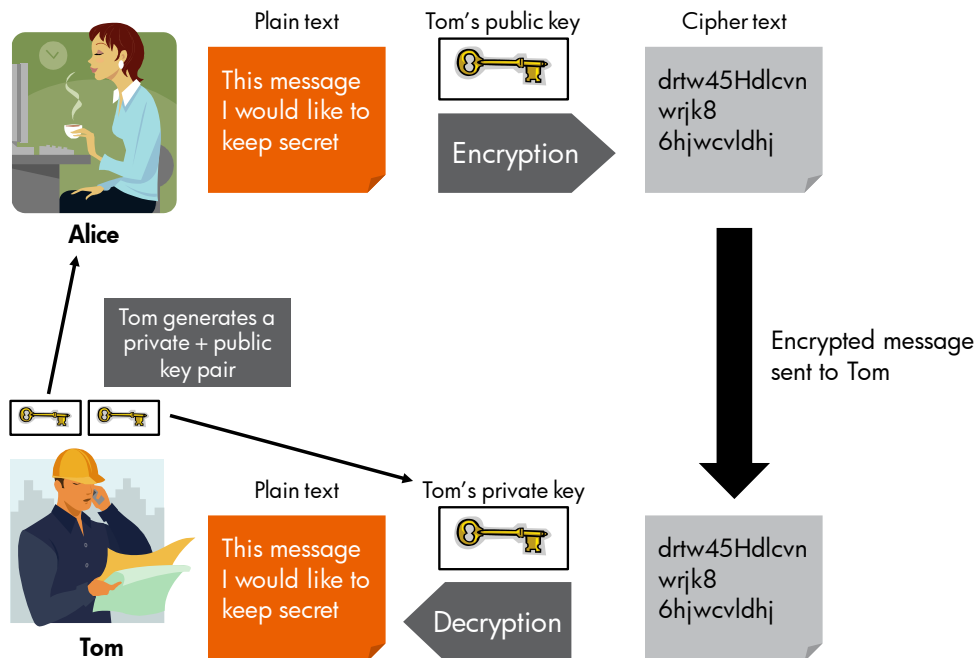
This technique solves key distribution problems and enables a Public Key Infrastructure (PKI). To explain public key/asymmetric encryption let's take an analogy, suppose Alice again wants to send Bob a secret message. Alice puts the message in a small box and padlocks it. The box is sent to Bob, but Bob needs the key from Alice to open it and Alice needs to get the key to Bob without it getting lost or stolen. This is the problem discussed in the preceding section on symmetric or secret key encryption.

This time imagine that the box is fitted with two padlocks. Bob has added his own padlock and kept the key. He then sends the box back to Alice. Alice can now unlock and remove her padlock, but cannot open the box as Bob's padlock is keeping it secure. She now sends the box back to Bob who unlocks his padlock and reads the message. This analogy shows that it is possible to send a secret message without sending keys and it inspired three cryptographers Diffie, Hellman, and Merkle to develop a solution to the key exchange problem.

Diffie, Hellman, and Merkle's new type of asymmetric cipher introduced the concept of private and public keys. However, they really only showed that a solution to key distribution is possible. The final practical solution to public key cryptography was developed by the three MIT university cryptographers: Ron Rivest, Adi Shamir, and Leonard Adleman. Their system, known as RSA, enables secret messages to be sent without key exchange. The Figure 3, shows how to implement an asymmetric cipher.

Alice wants to send Tom a secret message. Tom generates a public key and private key and keeps his private key to himself, but sends the public key to Alice. Alice encrypts the message using Tom's public key and sends the message to Tom. The nature of the cipher is that only Tom's private key decrypts the message and only public keys are sent over unsecured links. Obtaining the public key does not enable anyone to read messages.

Figure 3: Asymmetric or public key encryption



The cryptographic algorithm used in the RSA cipher is what is known as a one-way function using modular arithmetic and large prime numbers. Appendix C shows simple modular arithmetic. The algorithm also uses the fact that it is easy to multiply two prime numbers to produce a result. However, to undo the process and produce two prime factors from a given number is very difficult and time consuming for extremely large numbers. Essentially, the public key is the sum of the primes and is used to encrypt, but the private key requires the prime factors. By using prime numbers as large as 10 raised to powers in excess of 300 makes the encryption unbreakable as the search for the two prime factors would take all of the current world population of computers longer than the age of the universe. RSA was later founded as a commercial company.

Although effective, this type of encryption is slow and so well suited to low volumes of data such as electronic file signatures.

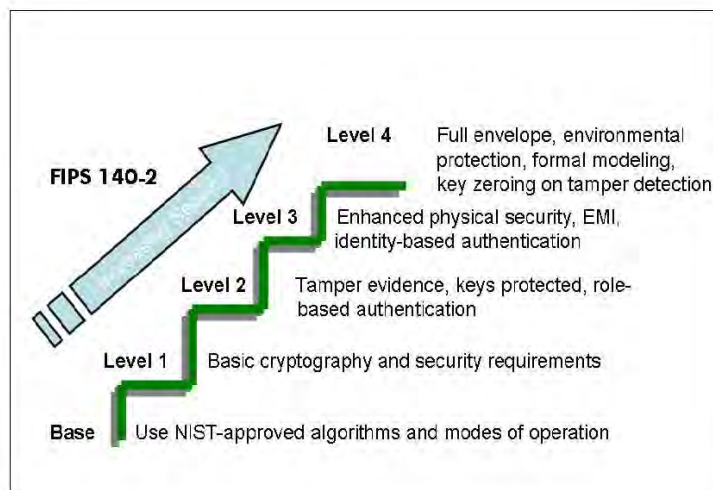
Encryption standards

Most cryptographic algorithms conform to specific U.S. and International standards published by a number of standards bodies. The primary standards organizations include: National Institute of Standards and Technology (NIST), International Standards Organization (ISO), and Institute of Electronic and Electronic Engineers (IEEE). In addition to these organizations the European Union (EU), has published a draft directive on digital signatures and encryption.

Some hash functions (message digests) are already based on approved standards, while others are awaiting standard verification such as those described in IEEE draft standard 1619.1. The LTO Ultrium Tape Drive employs the Advanced Encryption Standard (AES), also known as Rijndael. Already widely analyzed by mathematicians wishing to check its integrity, this block cipher algorithm has become a worldwide standard replacing its predecessor the Data Encryption Standard.

NIST is a U.S. non-government standards body and defines cryptographic standards in the Federal Information Processing Standards (FIPS) 140-2 document. AES is a FIPS-approved algorithm. FIPS 140-2 defines five levels of security for cryptographic modules, as shown in Figure 4 below.

Figure 4: FIPS 140-2 certification compliance

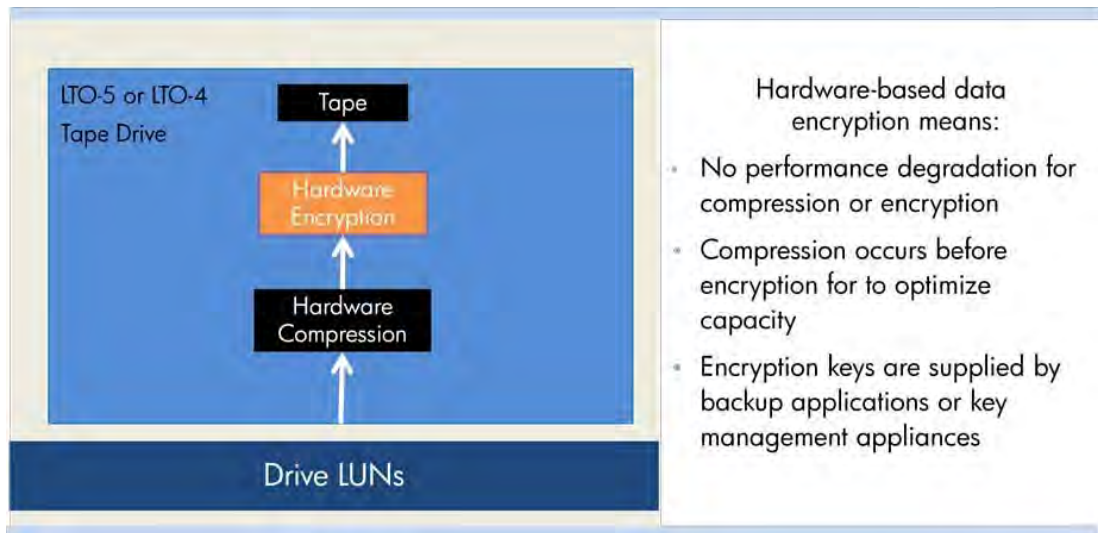


The LTO-4 and LTO-5 Ultrium Tape Drives have been awarded level-1 compliance with the FIPS 140-2 standard.

HP StorageWorks LTO Ultrium Tape Drive encryption

The LTO-4 format introduced the capability to encrypt (and decrypt) data within the tape drive hardware, and this is now also available with LTO-5 drives. This capability eliminates the need for software-based encryption and its inherent performance overheads (described on page 4). In addition to performance gains, tape drive hardware-based data encryption also improves the efficient use of available storage capacity through compression. Other methods of encryption leave compression until after the encryption process has taken place, often producing random data that cannot be compressed. The LTO Ultrium Tape Drive allows data to be encrypted following compression maintaining optimum storage efficiency.

Figure 5: Hardware-based data encryption



LTO encryption and interchange

Encryption is a standard part of the LTO-4 and LTO-5 format which requires that all drives must be “encryption aware”, which means that all LTO-4 and LTO-5 Ultrium tape drives from any vendor will return the appropriate sense codes when presented with an encrypted LTO cartridge tape. Implementing the encryption capability is, however, optional and consequently some manufacturer’s LTO-4 or LTO-5 drives may not have this capability.

Where drives have encryption enabled, interchange of encrypted data is made possible by the standard nature of the format specification, regardless of manufacturer.

The following table describes media support and interchange.

Table 1: Media support for encryption

Media	LTO-4 Tape Drives	LTO-5 Tape Drives
LTO-2 format tape media	Read only No encryption support	Not supported
LTO-3 format tape media	Read and Write No encryption support	Read only No encryption support
LTO-4 format tape media	Read and Write Encryption enabled	Read and Write Encryption enabled
LTO-5 format tape media	Not supported	Read and Write Encryption enabled

Key security with LTO

The LTO-4 and LTO-5 Ultrium Tape Drive encryption standard is AES Galois Counter Mode with a 256-bit key (please refer to the [appendix](#) for more detail on AES encryption). This is a secret key (or symmetric) algorithm, requiring the same key encrypt and decrypt data. To maintain security the key is not transferred to the tape cartridge under any circumstances and is only retained by the drive while power is retained, otherwise a new key is selected. Keys are supplied using the SPOUT SCSI command. Typically, a new key would be provided for a backup session, or for each tape.

LTO-4 and LTO-5 both use AES 256-bit encryption within the drive hardware. However, the new LTO-5 introduces the concept of “wrapped keys,” an NIST approved process intended to enhance the security for large quantities of data being handled. While the key is served in the same way (using the T10 SCSI command SPOUT), the LTO-5 drive encrypts using a random key generator within the drive. The key is re-generated every 232 records. This internal key is encrypted by the served key and then stored securely on the tape and the original supplied “served” key is required to “unwrap” or decrypt this key to read back the data. Effectively this means that every tape is encrypted with a different key even if the “served” key supplied by the ISV application does not change.

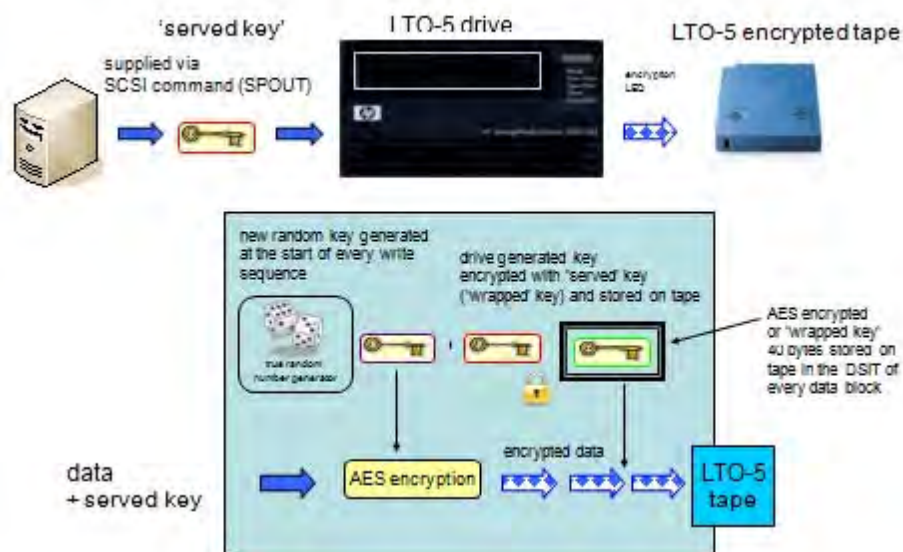
Should the drive be used to write an LTO-4 tape the data is encrypted as before using the “served” key provided by the SCSI “SPOUT” command. The new random number generator is used to provide initialization vectors in addition to the internal keys in LTO-5 format.

Reading back encrypted tapes in either LTO-4 or LTO-5 tapes still requires a key provided by the server (or key management system in library). Key labels can still be stored embedded in the format as authenticated data.

While reading encrypted data, the correct key must be supplied or a check condition is returned and the subsequent status indicates that either the wrong key has been supplied or to notify the user that the data on tape is encrypted (for example, if decrypt has not been selected).

LTO-5 drives have a blue encryption LED included on the front panel which illuminated when encryption is enabled. If there is an encryption/decryption error (such as incorrect key) then the LED will flash.

Figure 6: LTO-5 Ultrium Tape Drive—wrapped key



Additional security features

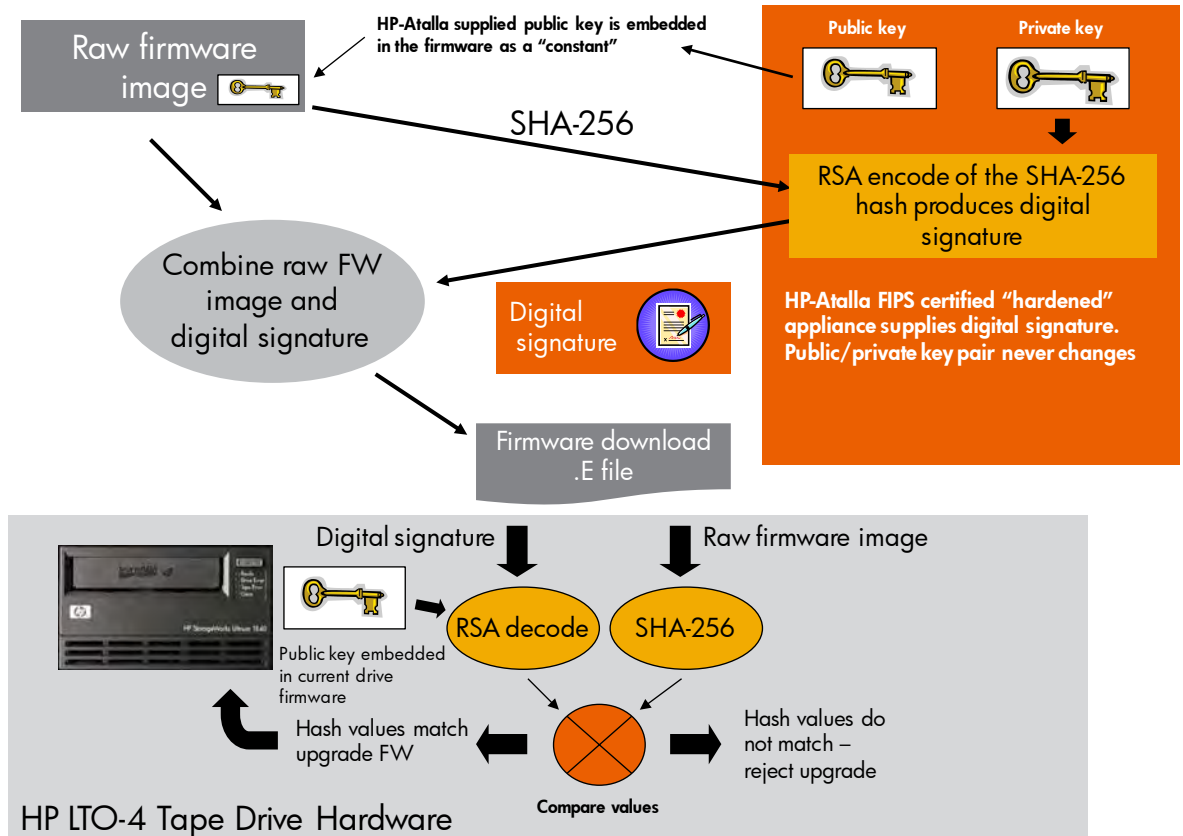
Digitally “signed” firmware

To achieve FIPS compliance, cryptographic devices are required to have digitally signed firmware. This ensures that it is not possible for the development of rogue firmware images containing embedded features that might compromise the encryption integrity. The HP StorageWorks LTO Ultrium Tape Drive achieves this by using an HP Atalla¹ secure appliance to provide digital signatures for the firmware image. This hardened appliance generates a Public/Private key pair. The public key is provided to HP, while the private key is held within the appliance. The appliance is FIPS certified to level 3 and any attempts to tamper with it result in self-destruction of the data held within. HP embeds the public key within firmware images and then performs a SHA-56 function on the data. [See the earlier section in this white paper on hashing \(digests\) and public key \(asymmetric\) encryption.](#)

The HP Atalla appliance then provides a digital signature, using RSA encryption with the private key, with each final firmware download file. While attempting a download of new firmware, by way of the drive SCSI/FC/SAS interface or tape, the drive runs a SHA-256 function on the firmware image. The public key held within the drive is used to perform an RSA decode on the digital signature, which contains the hash value expected from that image. The two hash values are compared and if not identical, the upgrade is rejected. If the firmware image was unauthorized, it would be impossible to load it. The following figure shows the process. Notice that the RSA decrypt uses the public key embedded within the existing drive firmware. This prevents an attempt to construct rogue firmware that would contain a key manufactured to result in a matching hash which if uploaded into the tape drive would compromise the encryption security. For example a rogue firmware could if loaded result in a tape drive which appears as normal to a host and responds to encryption enable commands but writes normal tapes. (Note LTO-5 drives use the HP corporate digital signature and not the HP Atalla secure appliance).

¹ Atalla is wholly owned subsidiary of HP specializing in electronic data security. Most ATMs in the world feature HP Atalla technology and the founder Dr. Martin M. (John) Atalla is often referred to as “father of the PIN”

Figure 7: LTO Ultrium Tape Drive firmware digital signing process



Prevention from reading raw data blocks

It is permissible within the LTO-4 and LTO-5 format to read raw data blocks from an encrypted tape, but without the key the data would simply remain as cipher text and make no sense. However access to raw data does make it possible to mount a "brute force attack" by attempting to decode the data using large amounts of computer power and a sequence of computer generated keys. To prevent this happening there is a security setting in the SPOUT SCSI command which controls "raw reads" of the data after it has been written to tape, by default "raw reads" are prevented and any attempt at a "raw read" returns a check condition.

Copying encrypted tapes

The combination of external write command and ability to read raw data makes it possible to copy an encrypted tape without being in possession of the key. This is seen as a security risk by HP and is not recommended. By default the HP LTO tape drives will report an error if a block of data is written with "external mode" set which is done via the "Security Protocol Out" SCSI command).

Multiple use of an incorrect key

The HP LTO-4 and LTO-5 drives prevent multiple attempts to read data with the incorrect key set. After ten attempts the drive will return a check condition until a new key is written or the drive is reset. This would make it a slow process to try out multiple keys against a block of data.

Practical use of HP LTO Ultrium Tape Drives with encryption

To use the encryption feature of the LTO-4 and LTO-5 Ultrium Tape Drives, you must instruct the tape drive to encrypt or decrypt data and issue the appropriate key. When power is removed, encryption is not enabled by default and the keys are not stored in the drive. The new SCSI commands Security Protocol In (SPIN) and Security Protocol Out (SPOUT) are used to set encryption and supply the key associated data, which is used to reference the correct key when restoring data. The appendix shows a screenshot from an engineering tool describing what is set using a SPOUT SCSI command. There are also two other RS422 serial interfaces on the LTO Ultrium Tape Drive. One is used for automation control in tape library applications. These additional interfaces can be used to set encryption parameters, load keys and check encryption status.

Note:

There is a third serial port, but only for diagnostic use.

You can implement encryption for tape drives in several ways and using different methods of key management. The following lists the different methods for completeness, however not all these methods are available as HP solutions, (or referenced solutions).

- Native mode encryption (sometimes referred to as “Set and Forget”). This method controls the LTO encryption from within the tape drive library. There is one key that is set by way of the library management interface (Web GUO or Operator Control Panel). This method encrypts all tapes with the same key, with the downside of negatively impacting the security level.
- Software-based encryption encrypts the data before it leaves the server and keys are stored in the internal database or catalog of the application. This method of encryption places a high load on the server as the software performs many mathematical operations using host processing power. Several applications including HP Data Protector software offer encryption as a feature. Although the security of data encrypted this way is very high (as the data is encrypted in transit), because encrypted data is highly random it then becomes impossible to achieve any data compression downstream in the tape drive and therefore storage is inefficient.
- Keys managed by the ISV application, also known as in-band key management. The ISV software supplies the keys and manages them, and the LTO Ultrium Tape Drive then performs the encryption. Keys would be referenced by the key-associated data and stored in the applications internal database. (Please refer to your individual ISV backup application vendor for support of this functionality).
- HP offers the StorageWorks 1/8 G2 and MSL LTO-4 Encryption Kit, providing an easy and affordable library-enabled solution for small businesses. One encryption kit is needed per library and includes two USB key server tokens. The key server token uses the USB port in the library and will generate and maintain encryption keys for the LTO-4 drives/libraries. The keys are transferred securely token-to-token for backup or export, with no exposure to insecure PCs, servers, or networks. The encryption kit is a self-contained solution for MSL libraries with no additional software, PCs, or servers required or involved.
- An in-band encryption appliance intercepts the Fibre Channel links and encrypts the data in-flight. These products are available from several vendors such as Neoscale and Decru. Key management is from a hardened key management appliance. This method is independent of ISV software and supports legacy tape drives and libraries. Data compression must be performed by these devices as compression within the tape drive is not possible after encryption.
- A SAN fabric switch with encryption capability is similar to the in-band appliance, but encryption hardware is embedded in the switch.

- A Key Management Appliance works with enterprise class libraries such as the HP StorageWorks EML and ESL E-series libraries. The HP StorageWorks Secure Key Manager which automates key management and is a hardened server appliance delivering secure identity-based access, administration and logging with strong auditable security meeting the rigorous FIPS 140-2 security validation. Additionally, the Secure Key Manager provides reliable lifetime key archival with automatic multi-site key replication, and high availability clustering. Encryption clients may access the cluster using flexible path and node failover capabilities.

Figure 8: Encryption and key management techniques compared







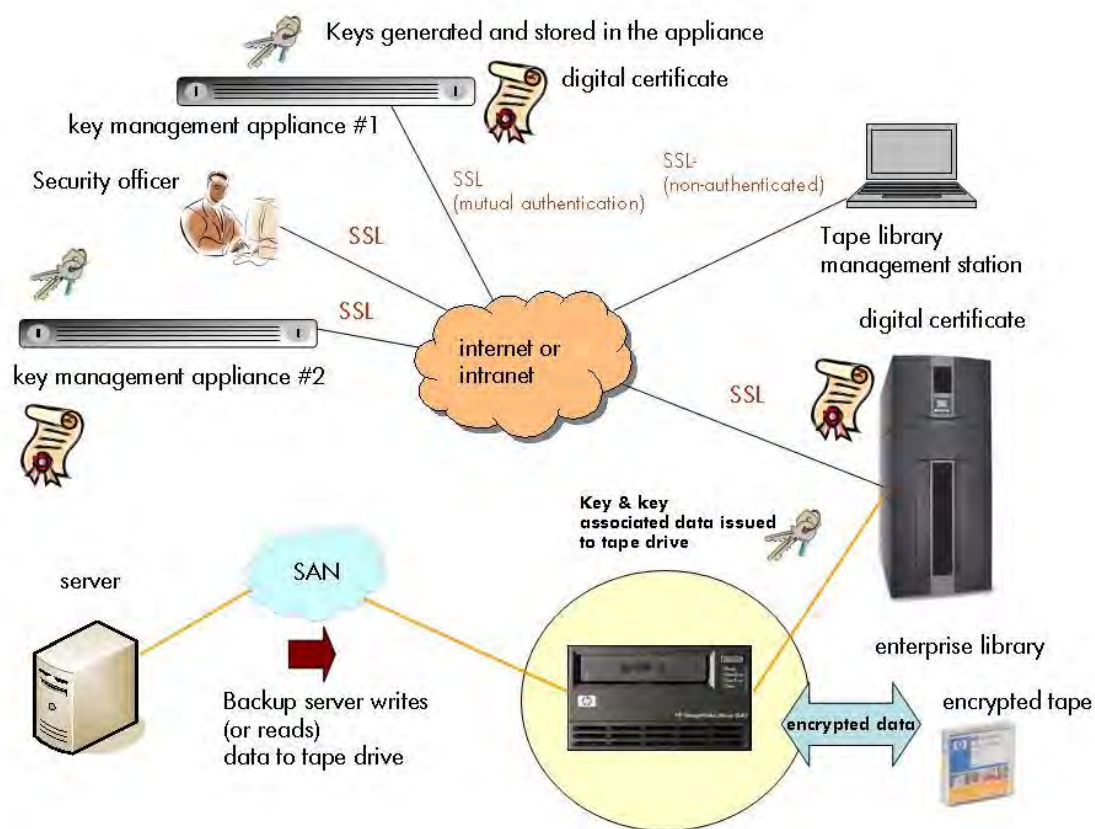
Encryption implementation	Advantages	Disadvantages	Notes	Cost	Security level
Native mode (Set and forget)—sets one key via the application or tape library	Inexpensive	Weakens security, manual key management.	Library vendor dependent. LTO4 support only	\$	
Software based encryption within ISV application	Supports legacy tape products. Data encrypted before it leaves the server	Degrades throughput. Tape drive data compression impossible		\$\$	
Keys managed via ISV application but encryption is LTO4 based	Good key management. Easy to implement	No legacy tape drive support	SMB software uses “passphrase” system	\$\$	
In band encryption appliance	ISV software agnostic. Supports legacy tape drives.	Expensive. Typically 2 – 4 drives per appliance	Additional key management hardware optional	\$\$\$\$\$	
SAN fabric switch with encryption capability	ISV software agnostic. Supports legacy tape drives	Requires additional key management.	Additional key management hardware required	\$\$\$	
Key management appliance	Very good key management, secure. Can be deployed as a cluster over WAN. ISV software agnostic	Expensive. Requires tight integration with tape library hardware. No legacy tape support	Enterprise class solution. Future deployment to other encryption products	\$\$\$\$	

Figure 9: Key management appliances and an Enterprise tape library



Notes on key management

Key management is a vital component of any cryptographic system. Keys must be generated, stored, and issued as required, but destroyed when no longer required.

Keys for the HP StorageWorks LTO Ultrium Tape Drive encryption function are 256 bits long with new keys typically issued for each tape. The SCSI initiator sets or unsets the keys and to accommodate multiple SCSI initiators, which are common in an enterprise-level application, the LTO tape drive can hold up to 32 different keys. Good practice encryption techniques require the generation of unpredictable random keys and realistically this is not a manual task.

Some applications use a passphrase system to generate keys, but this can lead to weakening the cryptography. Passphrase is generated by hashing the phrase with a secret number. However, hashes can be broken if guesses are made for standard English words or names. Modern computer hardware, for example, can break passwords which are produced by a hash algorithm in approximately 15 seconds if standard words are contained in the original password. However, passphrase generation can still be an effective solution in the SMB market where security of tape is important but a full key management system is expensive and too complicated. It is also necessary to have a key destruction system for when a tape is no longer in use or recycled by the backup application. In an enterprise wide key management unit there may be several thousands of keys in use at any one time.

It is also necessary to have a key destruction system for when a tape is no longer in use or recycled by the backup application. In an enterprise wide key management unit there may be several thousands of keys in use at any one time.

Warning

If the key for an encrypted tape is lost, then that tape cannot be read under any circumstances. AES-256 encryption is extremely strong and has not been broken to date.

Conclusion

Cryptography is an extensive subject; this white paper has been written to introduce the basic cryptography ideas and functions providing a greater insight into a practical data protection solution based on the HP StorageWorks LTO-4 and LTO-5 Ultrium Tape Drives.

An understanding of cryptography helps to provide a level of confidence in the security of the encryption used together with the importance of good key management as losing the key equates to losing data with tapes no longer accessible.

Standards are important in data protection and enable customers to meet increasing demands for legal compliance by demonstrating that sensitive data is adequately protected. Having industry-standard AES encryption as part of the LTO-4 and LTO-5 format adds further to the benefits of tape-based backup and archival, tape is now the most economical and one of the most secure forms of archival storage for valuable data.

The HP StorageWorks LTO-4 and LTO-5 Ultrium Tape Drives deliver both the performance and security features necessary to support the most robust data protection strategy.

Appendix A. AES encryption

AES encryption is asymmetric and uses a secret key. It is suitable for block mode encryption and has optional key lengths of 128, 192, and 256 bits. It operates on 16 bytes of data at a time and is arranged in a 4-byte x 4-byte array. The cipher was invented by Joan Daemen and Vincent Rijmen and was the winner of a competition run by NIST in the year 2000 to replace DES, also known as a block cipher. The full specification is available at: <http://csrc.nist.gov/publications/PubsFIPS.html> (FIPS standard publication 197).

Block mode ciphers, such as AES, process the data block in a set of mathematical operations called rounds. The key length determines the number of rounds. In LTO-4 or LTO-5 encryption, a 256-bit key is specified, which dictates 13 rounds. The process is reversible so that decryption is the reverse of encryption. As well as mathematically manipulating the data, a substitution takes place to add non-linearity. This is known as the S-box and is a fixed substitution. Used in combination with the byte re-ordering function, a very secure cipher is created. Full explanation of the AES cipher is beyond the scope of this white paper and is well described on many public Web sites.

AES only defines the encryption of one 4-byte x 4-byte array of data using a single key. To design a more practical solution means there are several modes of operation. In the LTO-4 and LTO-5 standard, Galois Counter Mode is used. This accommodates high-speed operation and is well suited for the 128-MB data rate of the LTO Ultrium Tape Drive. Counter mode operates by seeding a counter with an random number called the initialization vector (IV). This is incremented by one and the output is subjected to the AES encryption algorithm. This provides a stream of encrypted data, which is then combined with the real data using an Exclusive OR (XOR) function. The IV is reset at each record boundary and is recorded in the tape format. This is so that on read, the counter can be reset by the IV for that record. Galois-field mathematics is used to provide authenticated encryption. This is how the TAG value is computed. This provides additional security of the data record and the AAD, which can be used as a reference to retrieve keys from an appliance or from within ISV backup applications.

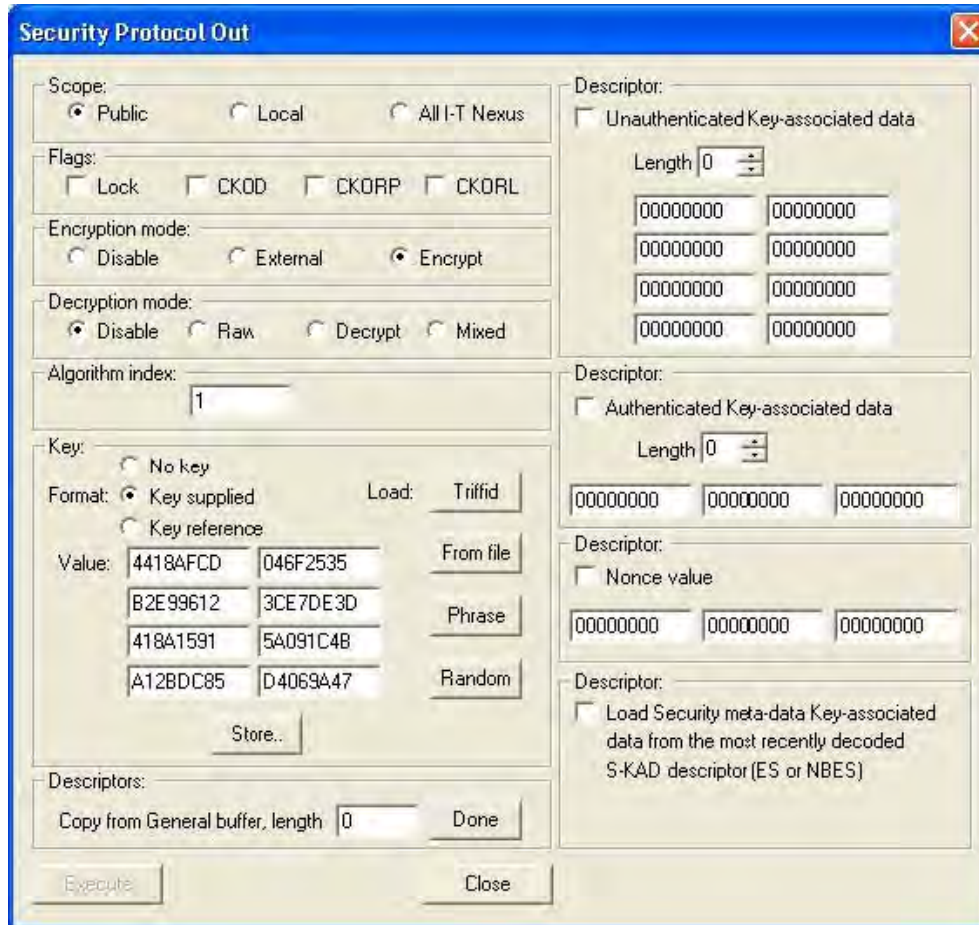
Figure 10 shows the typical parameters available in a SPOUT command. Notice the 256-bit key and the AKAD.

Appendix B. SPOUT parameters

Figure 10 shows some of the parameters which can be set by the SPOUT command. Note that the 256 bit key is shown and also the field for the key associated data. (The screenshot is from an engineering tool as this displays settings in a more visual format). These parameters would be issued by either the backup application for in-band management or the library management in the case of out-of-band management. The SPIN SCSI command would be used to obtain the encryption status of the drive and would return key associated data if required. Scope is used by each initiator to specify how the key is obtained. To understand this process, consider three different SCSI initiators A, B, and C, each accessing the drive in turn, just as if a backup application was sharing a drive between three servers.

Server A sets the scope to Local and issues a key of value X. When server A then writes, the data is encrypted using key X. Server B then issued a SPOUT command with the scope set at All_IT_Nexus with key Y. Server B then writes data encrypted with key Y. Server C then issues a SPOUT before its read with a scope of public. Server C does not supply a key, therefore the key supplied by the initiator setting ALL_IT_nexus flag is used. Only one initiator can set the All_IT_nexus scope. Setting a scope of Public resets the present key. If no All_IT_nexus scopy key has been set the data is written unencrypted.

Figure 10: Security Protocol Out screenshot



Appendix C. Modular arithmetic

Modular arithmetic is for integers only and has no carry function. Numbers just wrap around when they reach a certain value or modulus. It is similar to clock arithmetic used for working out what the time will be when you add a number of hours to a specific time to find a new time. For example, if a train leaves at 22:00 hrs and the next train is three hours later, what time does that train leave? The answer is 01:00 hrs in clock arithmetic, but in normal arithmetic $22 + 3 = 25$. Written mathematically, the sum is $22 + 3 = 1$ (modular 24). Notice that for standard AM/PM arithmetic, modular 12 is used and not 24. Reversing modular arithmetic is more difficult for larger numbers. For really large numbers the task is impossible. Modular arithmetic therefore produces what is known as a one-way function since multiple sets of inputs all produce the same result. For example: $22 + 3$, $22 + 27$, $18 + 7$ all yield 1 (mod 24) as a result.

Glossary

AAD	Additional Authenticated Data
AES	Advanced Encryption Standard
AKAD	Authenticated Key-associated Data
ALDC	Advanced Lossless Data Compression
ASIC	Application Specific Integrated Circuit
CA	Certifying Authority
CRC	Cyclical Redundancy Checksum
DES	Defense Encryption Standard
EOR	End of Record
Escrow	A legal arrangement for an asset (for example: encryption keys) to be held in trust by a 3rd party
EU	European Union
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
IEEE	Institute of Electronic and Electronic Engineers
ISO	International Standards Organization
ISV	Independent Software Vendor
IV	Initialization Vector
KMA	Key Management Appliance
NIST	National Institute of Standards and Technology
Nonce	Value used once—cf initialization vector
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
Served Key	Provided by key management device or server. In LTO-5 format embedded in the format. This is used to wrap the internal drive key before recording on tape.
SPIN	Security Protocol In
SPOUT	Security Protocol Out
SSL	Secure Sockets Layer
TLS	Transport Layer Security
VPN	Virtual Private Network