WHITE PAPER

ENTERPRISE ENCRYPTION

It's critical to understand how a defense-in-depth strategy can protect the most precious information asset: data.

Executive Summary

Organizations run on information, and for many, information is their most valuable asset. Keeping information secure in a highly connected and highly mobile world has become a top priority. Encryption of information is commonly cited as a proactive defensive measure.

By encrypting information, an organization has greater defenses against loss of sensitive data. Encryption strengthens the security posture and is often required by regulatory, privacy and compliance regimes.

However, encryption by its very nature can create challenges. Entities with mixes of public and private networks, virtual private networks (VPNs), applications, load balancers and layers of servers – which is to say most – can easily miss spots where encryption should be applied or misuse encryption.

Taking the time to define an encryption strategy gives IT administrators a clear expectation of what their roles and responsibilities are. By classifying sensitive or regulatory-driven data, security-killing ambiguity can be avoided, providing the best defense in depth and strongest security posture available.

TWEET THIS!

Table of Contents

What Is Encryption?

Before diving into when to use encryption, it's useful to have a brief review of the most significant terms and concepts in encryption, especially as they pertain to enterprise security. Because many of the algorithms in encryption have changed over the past few years, security experts who have not focused on encryption may want to verify that they are up to date with algorithms and concepts.

In the enterprise context, the term "encryption" broadly includes at least five specific technologies:

- Key exchange algorithms
- Symmetric (private) encryption algorithms
- Asymmetric (public) encryption algorithms
- Hashing algorithms
- Digital certificates and public-key infrastructure (PKI)

Although the specific algorithms in each change from time to time, the core ideas remain the same.

Best Practices Table

Area	Encourage	Discourage
Key management	Well-understood key management systems	Short personal identification numbers (PINs) for protecting long keys
Key exchange	Diffie–Hellman key establishment protocols	One-sided key establishment, as commonly used in Secure Sockets Layer (SSL) and Transport Layer Security (TLS)
Symmetric encryption	Advanced Encryption Standard (AES) encryption	Data Encryption Standard (DES), 3DES encryption, RC4 encryption
Symmetric key sizes	256-bit AES keys	Effective key sizes shorter than 128-bits
Asymmetric cryptography	Rivest-Shamir-Adleman (RSA) algorithm with key sizes greater than 1,023 bits and greater than 2,047 bits for long-term critical keys	RSA with key sizes shorter than 1,024 bits
Elliptic curve cryptography (ECC)	Use of ECC when mobility is important	For signing and key agreement if there is no specific mobility or performance requirement
Hash algorithms	Secure Hash Algorithm–2 using SHA-256 or SHA-512	Message-Digest Algorithm or MD5 (strongly) and SHA-1 (where possible to switch to SHA-2)
Wireless security	Wi-Fi Protected Access 2 (WPA2) with AES	WPA, non-AES encryption and all predecessor systems

An end-to-end encryption strategy takes the following steps:

- Sets both minimum and maximum requirements for what data should be encrypted, when it should be encrypted and how it should be encrypted
- Describes how keys are managed and controlled, and how key disclosure or loss is handled
- Defines specific situations in which data does not need to be encrypted, or should not be encrypted
- Covers common operational procedures (such as backups) and exceptional operational procedures (such as disaster recovery), along with associated policies
- Provides guidelines for encryption and hashing algorithms, certificate management, key lengths and other cryptographic parameters

Key Exchange and Management

Most encryption of data is done using symmetric (private key) encryption, meaning that both the sender and receiver of the data must have the same key in hand.

When encrypting data at rest (such as in a database or in a file system), a single long-lived encryption key has to be created and managed in some way. Where and how the key is stored varies dramatically from product to product.

In many cases, the key is "locked" using a PIN-type mechanism, such as a password or even a four-digit number. When evaluating key management systems, the strength of the key can be compromised if the key management system is not well designed or does not require a long enough key itself.

For example, if a 256-bit key is protected in a key management system with a four-digit PIN, the encryption could be compromised by someone guessing the four-digit PIN or simply brute-forcing the PIN (trying each of the 10,000 combinations). In some cases, such as when using smartcards, the key management system is hidden in some self-protective hardware. This can reduce the likelihood of a brute-force attack on the key management system, because the self-protective hardware might destroy its copy of the key after some number of incorrect attempts.

Although most "data at rest" key management systems are easy to understand and evaluate, "data in motion" key exchange is more opaque to application managers. Indeed, most network and application managers are unaware how their keys are exchanged, simply hiding behind the hope that Secure Sockets Layer (SSL) or Transport Layer Security (TLS) — in the case of network traffic — will magically keep data from prying eyes.

The best practice for all key exchanges is to require that a Diffie–Hellman algorithm be used for key management and exchange for any network application. The IP Security (IPSec) protocol, used for most site-to-site and some remote access VPNs, does this by default. But the very popular SSL and TLS protocols do not.

In fact, key exchange is one of the dirty little secrets of SSL and has led to fairly spectacular security failures in the past. For instance, one of the most popular web browsers selected poorly chosen random number generators to create encryption keys, making a brute-force attack simple.

Generally, key exchange in SSL (and TLS) is chosen by the server based on a list of options offered by the client. Security experts generally classify one-way key exchanges, in which one party makes up a key and sends it (protected) to the other party, as much less secure than two-way key exchanges, in which both parties contribute to the randomness and entropy of the key.

Even when the client is not at fault for offering (and the server for selecting) a less-secure key exchange, this mechanism in SSL/TLS allows a third party to attack the security of the connection. This is not by faking up a certificate, but by modifying the SSL/TLS negotiation to select poor security options that are easily cracked.

The default for most browsers when interacting with Microsoft application servers (such as Microsoft's Internet Information Services) is to select insecure (one-way) key exchanges, unless specifically configured to prohibit this. Although Microsoft, since Windows NT days, has made it possible to modify the allowed set of key exchanges, doing so requires customization of keys within the registry, far off the beaten path for most Microsoft server managers.

Unix-based servers using OpenSSL are more variable in what they will select, depending on the security-consciousness of the web server's developer. Application managers should edit their OpenSSL configuration files to block less secure key exchange methods.

To help identify potentially weak keys, network managers can use some firewalls and most intrusion protection system tools to alert, or even block, poorly chosen cipher suites.

Brute Forcing PINs

In 2012, DataGenetics' President Nick Berry published his analysis of four-digit PINs based on the many password files that have been leaked over the years, looking in detail at 3.4 million four-digit passwords. He found that a mere 20 PINs account for about 27 percent of the four-digit passwords: 1234, 1212, 1004, 2000, 6969, 1122, 1313, 4321, 2001, 1010 and the 10 PINs in the form 0000, 1111 and so on. That means that "brute forcing" a PIN often doesn't require trying 10,000 four-digit combinations. Instead, about a third of the time, it requires simply trying these 20 possibilities.

Symmetric Encryption

Most bulk encryption — encryption over more than a few octets of data — is done using symmetric encryption. This is true for both data at rest and data in motion. Even in cases where the amount of data may be small, such as individual email messages, symmetric encryption is commonly used.

For example, in the PGP cryptosystem, which is most often used for encrypted and authenticated email, each email message is encrypted using a symmetric (private key) algorithm, and only the encryption key itself is protected (encrypted) using asymmetric (public-key) cryptography.

Most IT professionals have fairly good working knowledge of symmetric encryption. However, there are two important points to highlight: encryption choices change over time, and key length matters.

For many IT professionals, cryptography started with the U.S. Data Encryption Standard (DES), an IBM-developed and U.S.-standardized algorithm for encryption using a key size of 56 bits (64 bits of which are parity bits). DES is no longer used, as brute-force attacks on DES have been shown to succeed in less than 24 hours. This means that any old data encrypted with DES, or which was transmitted in the past and could have been captured by an attacker, is effectively unencrypted.

Although DES was not commonly used in commercial computing, a variation on DES has seen heavy use as the preferred encryption algorithm in most IPSec VPN deployments: 3DES (usually referred to as "triple-DES"), which has an effective 112-bit key length, has withstood the test of time and is still considered a secure encryption algorithm. However, in 2001, the U.S. government issued a new Federal Information Processing Standard (FIPS) that uses the Advanced Encryption Standard (AES) as a replacement for DES and 3DES.

AES supports different key lengths, but the common lengths are 128 bits (the minimum) and 256 bits (the maximum), with 192 bits occasionally selected as "halfway in between."

As a best practice, and to avoid any questions of negligence, network and application managers should deploy AES for all new encryption applications and VPN deployments, selecting the 256-bit key length wherever possible. However, for information that is sensitive for only a short period of time, the 128-bit key size is considered secure against contemporary attacks.

Another important algorithm to be aware of is RC4. A stream cipher, rather than a block cipher such as 3DES, RC4 is often selected for streaming applications. It is used in the original wireless encryption algorithm, Wired Equivalent Privacy or WEP (with a 40-bit key), as well as BitTorrent, SSL/TLS, Microsoft Remote Desktop Protocol (RDP) and Skype. Although RC4 can have long key lengths (thousands of bits in some cases), almost all applications based on RC4 use a length ranging from the minimum allowed, 40 bits to 256 bits.

Although RC4 is still heavily used, it is a poor choice for a couple of reasons. Because RC4 was used in WEP (the original encryption protocol with 802.11 Wi-Fi LANs), it has become synonymous with "poor security." Although WEP itself was at fault, not the RC4 algorithm, the association between WEP and RC4 has caused many security professionals to shy away from the algorithm, either from ignorance or simply to avoid having to explain why RC4 wasn't the reason that WEP was insecure.

Additionally, RC4 has come under a number of attacks from cryptographers. And several issues have been found that question the long-term viability of RC4 for enterprise use.

Neither of these is a reason to run screaming from RC4. But network and application managers should cross RC4 off their list for any future deployments and should be looking at places where RC4 might be in use — such as in SSL/TLS servers, where it is still common, especially in Microsoft environments.

Network managers will notice a wide variety of new encryption algorithms also popping up, such as Camellia, which was introduced and patented by NEC and has been selected by some application providers. Generally, unless a specific requirement is made to use a nonstandard algorithm, AES should be selected.

Its cryptographic properties have been more carefully studied than other, similar algorithms. In addition, Intel and, more recently, AMD have included AES acceleration in their CPUs, giving a significant performance increase for servers and helping to improve overall security by avoiding certain types of timing attacks that have been proposed against AES.

What Is Elliptic Curve Cryptography?

Elliptic curve cryptography is popping up in application and virtual private network (VPN) servers, driven by the increasing number of low-speed devices, such as smartphones and tablets, that are connecting to the Internet. ECC has been used to speed up both digital signatures and key agreement, two computationally intensive operations.

The main idea behind ECC is that smaller key sizes and simpler computations provide security equivalent to much larger key sizes of other algorithms. For example, a 3,072-bit RSA public-key computation can be replaced by a 256-bit ECC public-key computation with the same level of security.

Standards bodies have avoided use of ECC because some aspects of the cryptography are covered by patents owned by Certicom (a subsidiary of Research in Motion or RIM). But the cloud of uncertainty about public use of ECC has been partially lifted by a U.S. National Security Agency license of the patents for government use.

Network and application managers focused on supporting mobile devices that can benefit from ECC, especially in the U.S. government sector, should investigate the performance benefits that it can offer without compromising security.

Asymmetric Cryptography

Although encryption is the primary goal, many encryption systems depend on a combination of tools to accomplish other tasks. Public-key cryptography is one of those tools. Although public-key cryptography is rarely used for encryption of long strings of data because of its fairly slow performance, public keys are used heavily for signing messages (authentication and integrity checking) as well as encrypting short strings (such as session keys).

One algorithm is heavily used in public-key cryptography: Rivest-Shamir-Adleman (RSA), the original public-key cryptography algorithm from 1978. The Digital Signature Algorithm, or DSA, based on a 1984 algorithm developed by Egyptian cryptographer Taher Elgamal and used only for signing, not encrypting, is also widely available.

When network managers have a choice between the two, typically RSA is more widely supported — mainly because it can be more widely applied.

One important consideration for security-conscious managers is key size. RSA keys should be chosen based on the sensitivity of the information being protected and the expected lifetime of the sensitivity. Keys sizes of 512 bits are now considered insecure — one was "broken" in 1999 in seven months.

RSA (the company), through its RSA Laboratories, suggests that organizations select key sizes of 1,024 bits (considered about equal to an 80-bit symmetric encryption key) for ordinary enterprise use and 2,048 bits (similar in strength to a 112-bit symmetric encryption key) for extremely valuable data, such as certification authority root keys.

For information that must be protected for more than 20 years, the U.S. National Institute of Standards and Technology recommends a 3,072-bit RSA key, which is roughly equivalent in strength to a 128-bit symmetric encryption key.

An even more important area of attention when using RSA is key lifecycle. Most RSA (and DSA) keys are used many times over their lifetime; for instance, when incorporated into digital certificates. Enterprise security managers should strictly require that keys be replaced every few years. A maximum of three years is considered good practice.

Hashing Algorithms

Hashing is used to ensure that information is not modified in transit. Hashing is heavily used in all aspects of encryption, including in VPNs, digital signatures and certificates, and email encryption.

Hashing is often an afterthought for encryption applications, but improperly selected hash algorithms have resulted in some spectacular and very public failures. In the 1990s,

security researchers began to identify flaws in one of the two commonly used hashing algorithms, Message-Digest 5 (MD5). The U.S. government replaced MD5 with Secure Hash Algorithm-1 (SHA-1) in 1993 as a standard for federal agencies.

In 2005, SHA-1 was also shown to have internal cryptographic weaknesses. So security-conscious systems managers should avoid all use of SHA-1 and move to the newer SHA-2 algorithm. SHA-2 is actually a family of hash algorithms with different size digests, so these may be referred to in product documentation by their hash sizes: SHA-224 and SHA-256 (32-bit-wide algorithms) and SHA-384 and SHA-512 (64-bit). If 64-bit devices or servers are in place, SHA-512 will give substantially better performance over SHA-256 and is generally preferred.

Encryption and the Cloud

When an organization doesn't control the infrastructure, there's a huge question to answer: Will the organization's cloud service provider safeguard its data? Or is it up to the entity to do that?

It's critical to alleviate this concern through encryption use. Any cloud application, whether private, public or hybrid, must use an encrypted VPN for all communication. Encrypting data in transit solves some problems and is a clear requirement for any cloud-based application. But encrypting data in transit doesn't help secure data at rest.

Cloud applications have two significant risks that encryption can help mitigate. One risk is familiar: An application could have holes or bugs that let an unauthorized party view sensitive data. The other risk, more specific to cloud service providers, is that the infrastructure might not be secure.

Encrypting data in the app protects against both types of risk but may not be desirable. For example, cloud-based email, such as an outsourced Microsoft Exchange service, will easily support Secure/Multipurpose Internet Mail Extensions to encrypt sensitive mail on the server. But S/MIME presents a different set of problems, including interference with archiving and search, long-term key storage issues, and generally weak support in popular mobile and web-based email clients.

In some cases, encryption of data in the cloud is the easy and an obvious choice, such as with offsite storage for backup, file sharing or collaboration. In other cases, such as customer relationship management (CRM) or help desk applications, encryption of data likely won't work, and organizations must depend on the assurances, reputation, auditing and best practices of their cloud service provider.

Points of Encryption

Trying to make sense of the mishmash of encryption strategies can stress anyone, as overlapping options and a whole host of what-if scenarios make it difficult to design a holistic approach that doesn't over-encrypt or under-encrypt. Although every enterprise application and network environment is different, it's helpful to divide encryption into two main categories: protecting data in transit and protecting data at rest.

As a simple guideline, start by assuming that the two types of protection are not connected in any way: Data that is already encrypted on disk (at rest), for example, still needs protection when sent over the network (in motion). In other words, encryption strategies should assume that all data needs to be protected in motion, even if it's already been encrypted at some other layer.

This assumption has some key benefits:

- Data that is already encrypted gets additional protection, such as metadata hiding.
- Any error on the application layer in missing encryption is partially compensated for by encryption in motion.
- Disconnecting "at rest" and "in motion" encryption simplifies "what if" scenarios in environments where users connect using different networks.

Protecting Data in Motion

Network and application managers will find common tools for encryption at three different network layers: data link, network and application. These are briefly detailed in this chart:

Encryption Layer	Common Tools	Advantages	Disadvantages
Data link	Media Access Control (MAC) Security protocol (wired), WPA2 (wireless)	Low performance impact; low configuration cost	MACsec (802.1ae) not commonly supported
Network	IP Security (IPSec) VPN, SSL VPN	A combination of authentication, encryption and access control	Requires a client; has a performance impact; is not easy to use in LAN environments
Application	SSL/TLS, Secure Shell tunneling	Works for all users; generally does not require specialized client	Can impact server and client performance; often is not compatible with network-based data loss and intrusion prevention tools

Unfortunately, encrypting and protecting data at only one layer doesn't solve all problems. The result is that it's too easy to encrypt everything three (or more) times as it passes over a corporate network simply by using normal security tools, such as wireless Wi-Fi Protected Access 2 (WPA2) encryption, server-level SSL encryption and standard VPN products. The result is additional overhead and a performance hit, especially for remote users.

Network and security managers trying to reduce double (or triple) encryption often look at removing one of the encryption layers to increase performance and reduce complexity.

Generally, there is no reason to reconsider data-link layer encryption. Data link encryption for wireless networks doesn't actually create overhead or slow down traffic in any measurable way. In addition, no one would consider deploying an enterprise wireless network without using WPA2 (AES-based) security, which includes authentication, encryption and message integrity checking.

When providing direct access to encrypted application servers, host-based intrusion protection (including break-in evasion) is a clear necessity. Without it, a dedicated attacker can attempt a brute-force attack, which could go completely undetected or unprotected. Opening up application servers to direct connections also invites denial-of-service (DoS) attacks using commonly available tools.

When an application server is hidden behind a VPN concentrator, it is possible for a DoS attack to be launched against the VPN concentrator. But the two-step disassociation between the VPN concentrator and the actual application server leaves the application server protected, and also presents a less obvious target to an attacker.

The opposite approach leaves the network-layer VPN protection in place, but removes application-layer encryption. This strategy is more common among performance-conscious application owners because it retains the minimum required protections while offering the maximum performance.

The danger of this approach is that someone will make a configuration error somewhere, perhaps months or years after initial deployment, and the unprotected application server may be used in a way that does not offer encryption of data in motion. When corporate networks are very large, or when information is very sensitive, unencrypted data even over a LAN — normally considered to be secure in most organizations — may not be acceptable either.

There is no perfect way to avoid doubly encrypting traffic. But it is important that the application, network and security teams agree on an approach for most enterprise applications. By having a consistent view on how data in transit should be encrypted, the risk of expensive and embarrassing data breaches can be significantly reduced.

Protecting Data at Rest

Data at rest is encrypted for a variety of reasons, including regulatory compliance. However, the main focus of any encryption deployment should be threat mitigation.

This can be broken down into two subsets: user devices (smartphones, notebooks and desktops), and servers and all of their associated hardware (such as backup appliances).

User Systems

For phones, tablets, notebooks and desktops, the main threat is loss of physical control of the system: outright theft, misplaced or lost devices, "borrowed" devices and improper destruction of data when systems are recycled.

When selecting encryption solutions for notebooks and desktops, most organizations have the advantage of a sophisticated infrastructure. In contrast, mobile devices such as phones and tablets are usually handled by mobile device management (MDM) software. Many of the functional requirements for encryption of notebooks and desktops are impossible to meet in the world of mobile devices, so enterprise managers may need to mix and match solutions to cover all their users' systems.

The table below shows enterprise-level requirements for user device encryption. Because the requirement for encryption is now so common, all the desktop operating systems (recent versions of Windows as well as Mac OS X) and major mobile OSs (recent versions of Android as well as Apple iOS) all cover a significant subset of the requirements below as a part of core OS services. But none cover all of them with built-in encryption.

Area	Requirement
Encryption	Requires full-disk encryption; per-file and per-directory leaves devices vulnerable to attack
Authentication	Requires end-user authentication prior to boot and periodically thereafter; hardware token (such as smartcard or USB dongle) is a good two-factor addition
Key recovery	Requires enterprise management (including revocation and replacement) and backup of encryption keys; no device can be encrypted without a key managed by the organization
Removable media control	Requires control of removable writeable media (especially USB drives, but also CD/DVD writers); allow only for corporate-provided media meeting encryption requirements
Logging, audit and compliance	Requires standards certification (such as the Federal Information Processing Standard or FIPS 140), and full logging and auditing to demonstrate compliance with policies and regulatory regime
Enterprise infrastructure integration	Requires integration with enterprise directory product (such as Active Directory's Windows Domain) for user authentication and key management and control

Windows Flash Drive Encryption with BitLocker

Since the release of Windows 7, Microsoft has made it possible to extend BitLocker To Go encryption of disk volumes to USB volumes, such as portable thumb drives.

When a domain-connected Windows system has the appropriate group policy, any insertion of a removable volume can kick off BitLocker To Go tools that will enforce encryption of the entire volume. Windows 7 and above systems can read these encrypted devices automatically. For older versions of Windows, the BitLocker Reader application is available to decrypt a locked volume.

Protecting Data on Servers

Encrypting data on servers is a more complicated process than protecting user data for several reasons:

- Multiple-access file systems (such as clustered file systems) or network-based filing may not be whole-disk encryptable.
- The threat of loss of physical control is low, but the threat of an intruder gaining access to the system is much higher and needs to be mitigated differently.

For these reasons, most entities choose not to use full-disk encryption on servers and instead focus on other threat mitigation options including file- and folder-based encryption as well as endpoint security products, host-based intrusion prevention, integrity monitoring, and a heavy dose of security information and event management (SIEM) alerting on the logs from these systems.

Some types of data should always be encrypted, such as personally identifiable information (PII), protected health information (PHI) and or private financial information (such as credit card numbers).

Protecting such data by having applications encrypt and decrypt on the fly is a common strategy. Although experience has shown that application developers aren't necessarily good at selecting cryptography to protect sensitive information. A better approach is to store the information in a database and make use of one of the Transparent Data Encryption features of the database (or an add-on package that provides TDE).

TDE supports real-time encryption and decryption of data (and database transaction logs). Most TDE products — including those built into some database management systems — support hardware-based key protection. To optimize performance, these products often also allow the database manager to select columns, tables or entire databases for encryption.

As with any encryption product, TDE tends to create issues in scalability and high availability because keeping encrypted information synchronized across databases, or distributed across multiple systems, is more difficult than the same task with unencrypted information.

In large-scale applications, TDE encryption may not be suitable, making the use of application-layer encryption and decryption necessary. But application managers should focus on built-in encryption/decryption facilities, such as TDE, and avoid creating custom encryption services.

Another issue for server environments is backup data – although this also applies to user devices. Obviously, having unencrypted data on a backup tape that is beyond the organization's control, or is loosely controlled, is just as bad as having unencrypted data on a notebook in the trunk of someone's car. Therefore, even if the application and file servers themselves are not encrypted, enterprise managers should ensure that all backup data is encrypted.

Arrival of Windows 8

Windows 8 brings many of the same encryption features available in Windows 7 and earlier versions, with minor twists.

Core encryption technologies that have been present in other versions of Windows — such as the IPSec VPN, SSL/TLS in Internet Explorer, certificate services and cryptographic acceleration for compatible hardware — are all still very much present and should be familiar to desktop managers.

The biggest changes relevant to encryption have occurred in the relatively new BitLocker and BitLocker To Go (Microsoft's drive encryption solutions), and in Microsoft's sandboxing technology. As in earlier versions, not every edition of Windows 8 has the same feature set. For BitLocker, Microsoft's drive encryption solution, the Pro and Enterprise editions are required.

BitLocker and some associated application programming interfaces in Windows 8 have been updated based on customer requests for a more user-friendly encryption solution.

Because Windows 8 operates on tablet, notebook and desktop systems, and runs across multiple CPU architectures, some internal changes were made to broaden drive encryption capabilities. For example, self-encrypting hard drives (often called eDrives) and advanced format drives (those with 4,096-octet rather than 512-octet sector size) are better supported in initial releases of Windows 8. Note that the use of eDrives with Windows essentially delegates all trust and security of the drive to Microsoft.

Desktop managers may need basic input/output systems (BIOS) updates, firmware updates or even Windows patches to fully support BitLocker on all eDrives and advanced format drives. Unsolved issues with BitLocker include SkyDrive cloud storage encryption and interactions between BitLocker and SharePoint.

BitLocker Administration is still in beta test with Microsoft BitLocker Administration and Monitoring version 2 (MBAM 2.0). But key new features that Microsoft shipped in beta versions last November include a self-service portal to resolve some issues without having to call help desk support and compatibility between MBAM 2.0 and Microsoft System Center 2012 Configuration Manager. Other additions include changes to improve the end-user experience when installing and interacting with BitLocker and BitLocker To Go.

For a complete Windows 8 drive encryption solution, a v1.2 (or later) Trusted Platform Module chip is highly recommended. Although most devices being made today have TPM chips, some low-end consumer PCs do not. TPM is critical for proper storage of drive encryption keys and also for detecting certain low-level attacks on the Windows operating system. Keep in mind, some countries (for example China and Russia) have strict TPM regulations.

CDW: A Security Partner That Gets IT

Encryption refers to a comprehensive set of product offerings to protect data and information assets from multiple threats. CDW offers a wide selection of security solutions that protect data, both at rest and in transit.

Your CDW account manager and solution architects are ready to assist with every phase of choosing and leveraging the right encryption solution for your IT environment.

Our approach includes:

- An initial discovery session to understand your goals, requirements and budget
- An assessment review of your existing environment and definition of project requirements
- Detailed manufacturer evaluations, recommendations, future environment design and proof of concept
- Procurement, configuration and deployment of the final solution
- 24x7 telephone support, as well as ongoing product lifecycle support

Security Assessments – CDW security assessments are tailored to reflect organizational needs. Each security report highlights individual concerns and goals. CDW security assessments include analysis of any or all of the following:

- Internet security
- Internal network security
- Partner or extranet security
- Comprehensive assessment
- Dial-access security
- Wireless network security
- Data loss assessment

To learn more about CDW's security solutions, contact your CDW account manager, call 800.800.4239 or visit CDW.com/security

---

SOPHOS

Sophos Complete Security Suite gives you the antivirus, endpoint and mobile protection you need with the device control, encryption, web and email gateway security you demand. And, because it's all from Sophos, it works better together. It's backed by a vendor you trust. Even better, it's so simple to use you'll actually turn it on – delivering exceptional protection that saves you time and money.

CDW.com/sophos

McAfee® An Intel Company

McAfee® Endpoint Encryption™ solutions use powerful encryption algorithms and offer multiple layers of data protection that address specific risk areas. Encryption is extended to desktop PCs, notebooks, network files and folders, removable media and USB storage devices. Endpoint Encryption allows you to transparently secure a broader scope of confidential information including customer data, intellectual property, legal and financial records, and employee communications with no system performance degradation.

CDW.com/mcafee

Symantec.

Symantec's encryption solutions enable organizations to deliver data protection with centralized policy management through the optional use of encryption management server. Our solutions provide standards-based technology, centralized policy management, compliance-based reporting and universal management for your encryption products.

CDW.com/symantec

TWEET THIS!

CDW® PEOPLE WHO GET IT™