Tivoli software

IBM

Using IBM Tivoli Key Lifecycle Manager: Business Benefits and Architecture Overview

Redguides
for Business Leaders

Axel Buecker
David Crowther

- Business benefits and IT challenges in deploying a centrally managed key life cycle infrastructure
- Overview of the architectural components for IBM Tivoli Key Lifecycle Manager
- Real-world implementation experiences

Redbooks

Executive overview

These days the world is becoming more and more globally integrated. As individuals we can buy goods and services using an e-commerce Web site hosted in one country, with the goods being manufactured in and dispatched from another country, billing occurring in another, and technical support and accounting coming from still other countries. And yet these processes are integrated so as to provide us with a single view of the whole transaction. For businesses, this global integration provides benefits as well as challenges and problems. The integration of global financial systems can mean that a local issue in one country can quickly spread, becoming a global crisis.

As a consequence of this business globalization, new supporting technologies and new computing models are constantly emerging. The technical infrastructure for globalization is in part provided by emerging technologies, such as *cloud computing* (which enables access to data anywhere, at any time), *social networking* (which creates personal and business networks that can rapidly expand worldwide), and *real-time data streaming* (which provides updates on events as they happen). help provide the technical infrastructure for globalization. *Information technology* (IT) itself is being much more widely deployed and integrated to help us implement more sophisticated *business processes*. For example, surveillance systems are being integrated with automatic car license plate recognition systems, and in concert with mobile phone and credit card transaction data, law enforcement agencies are more able to quickly track down suspected criminals around the world. It is not just the consumer who is affected; *businesses* are experiencing more pressure from the global economy, such as the need for 24x7 systems to support a world-wide customer base, and competition from companies in developing countries who are now easily able to market their products around the globe. *Federal organizations* are under constant pressure to deliver more results with less resources and less time.

IBM® calls their initiative to monitor and advance this globally integrated technology *Building a Smarter Planet*™. In order to participate in this *Smart Planet*, organizations need to build and maintain a suitably *Smart Dynamic Infrastructure*—one that is flexible, provides high performance, and is cost-efficient and secure. In this IBM Redguide™ publication we take a closer look at one particular aspect of the Smart Infrastructure, namely *data security.*

Organizations have to effectively manage their data to ensure it is *available, secure*, *resilient,* and *compliant*. At the same time the volume of data is growing year over year, often at double-digit rates, increased threats to that data from within and outside the organization, increasing regulatory requirements from governments or oversight committees, as well as more stringent contractual requirements from business partners are all challenging the

organization. Data is a crucial resource of any organization, and if it is lost, compromised, or stolen, the effects on the business can be truly devastating. One of the ways organizations can fight threats to their data is by turning to *encryption* for the protection of their *data at-rest*[1].

However, encryption itself requires careful on-going management if the data is to remain available and secure. The encrypting software or hardware, or both, is typically chosen for, among other things, robustness and performance. The *encryption keys* play a pivotal role in this because if the key used to encrypt the data is lost or compromised, then the encrypted data itself is also compromised. As a consequence, the keys used to encrypt the data require ongoing (or life cycle) management to ensure they are also kept secure and are available when required without adding to the complexity of the overall solution.

As the amount of data grows exponentially, so too do the number of encryption keys. Each key may apply to one or more than one data device. Each key has a date after which it becomes invalid; new keys must be created when old ones expire or new devices are brought into the data environment. Keys must be provided to third parties or received from third parties to enable the exchange of data, and compliance, reporting, and auditing requirements all must be met. In short, the life cycle of each key, from creation to retirement and removal, poses a management challenge, which increases as the number of keys in use increases.

Today, key management is often fragmented: department teams may use manual processes or built-in tools, some keys may be centrally managed using secured databases, and in some cases, no key management exists at all. The lack of proper management can lead to loss or compromise of keys and the subsequent loss or compromise of encrypted data.

In this guide we describe the challenges of key management, the IBM solution to this complex problem, and why we moved to support hardware-based encryption in the storage infrastructure. We discuss how IBM Tivoli Key Lifecycle Manager (TKLM) can be integrated with your IT infrastructure to help you strengthen your organization's key management processes and keep control of a large and increasing key pool. We also show how Tivoli Key Lifecycle Manager can ensure your tape cartridges and disk drives are protected when they are moved in-house or off-site, eliminating the worry that lost storage media could damage your business or influence your compliance posture.

IBM Security Framework

Today's business leaders are expected to manage risk in their areas of responsibility in the same way that CFOs manage risks in their domains. Security risks and the potential impact on IT must be communicated to executive peers in business terms. Additionally, IT security controls must be aligned with an organization's business processes, IT risks monitored and quantified in business terms, and business-level insight into IT risks provided at the executive level. In other words, business leaders need to manage risk and orchestrate security operations in a way that enforces compliance and optimizes business results.

As an organization secures its business processes, a business-driven approach needs to become the guiding influence for ensuring that all the different security domains work together in a holistic and synergistic manner, in alignment with the overarching business objectives. Otherwise, the organization's risk stance becomes vulnerable due to misalignment of priorities between IT and the business strategy. Using a standards-based approach to map business drivers to IT security domains is often very difficult and is often an afterthought.

[1] *Data at rest* is considered data that is being held in any type of computer storage, for example, a disk or a tape library. When this data is being retrieved from storage it is considered *data in transit*. While the data is in transit it is typically moved between locations like local computer memory and networks of all sorts, including the Internet.

IBM created a comprehensive IT security framework that can help ensure that every necessary IT security domain is properly addressed when using a holistic approach to business-driven security. This framework is illustrated in Figure 1.

Figure 1 The IBM Security Framework

IBM provides a broad range of solutions and services that can enable organizations to take this business-driven, holistic approach to security in alignment with the *IBM Security Framework*.

Comprehensive *professional services*, *managed services*, as well as *hardware and software* offerings, are available from IBM to support your efforts in addressing the following *security domains* covered by the IBM Security Framework.

Security Governance, Risk Management, and Compliance

Every organization needs to define and communicate the principles and policies that guide the business strategy and business operation. In addition, every organization must evaluate its business and operational risks, and develop an enterprise security plan to serve as a benchmark for the execution and validation of the security management activities that are appropriate for their organization.

These principles and policies, the *enterprise security plan*, and the surrounding quality improvement process represent the enterprise Security Governance, Risk Management, and Compliance model. Within this overall model, the requirements and the compliance criteria for the remaining security domains are:

- People and Identity

  Assure that the right people have access to the right assets at the right time.

- Data and Information

  Protect critical data in transit or at rest across the organization.

- Application and Process

  Ensure application and business services security.
- Network, Server, and Endpoint (IT infrastructure)

  Stay ahead of emerging threats across IT system components.
- Physical Infrastructure

  Leverage the capability for digital controls to secure events—or people or things—in the physical space.

These domains are built on a foundation of due diligence in security governance, risk management, and compliance disciplines. The framework describes and explores the business areas that need to be addressed, but does not attempt to define *how* the security domains are actually implemented in hardware, software, and services.

While other security vendors can only address part of this whole picture or manage part of the risk, the domains of the IBM Security Framework represent IBM's view towards a holistic, business-driven approach for IT security. IBM continues to bring a strong focus and significant investment to the market in order to provide the right technologies and expertise to deliver leading-edge security solutions, from IT security asset life cycle management to operational control over security transactions, across every domain. In this way, IBM can provide unparalleled capability to secure complete business processes.

Now that we have introduced the IT security domains, it is time to look at the more technical aspects of creating the security architecture. In the following section, we discuss the *IBM Security Blueprint,* which can guide you and your IT security professionals to identify the architectural principles that are valid across all domains and environments, as well as the fundamental services within and across the domains and environments.

IBM Security Blueprint

The IBM Security Framework divides the area of business-oriented IT security into several domains. The next step is to break these down into further detail to develop an architectural framework that can help define and implement your organization's goals. This architectural framework is called the *IBM Security Blueprint*.

The IBM Security Blueprint uses a product-agnostic and solution-agnostic approach to categorize and define security capabilities and services that are needed to satisfy business security requirements or areas of concern categorized by the IBM Security Framework. It also defines a common vocabulary to use in further discussions.

In the blueprint, IBM aims to identify architectural principles that are valid across all domains, and fundamental services within and across the domains. The blueprint also highlights applicable best practices and IT standards.

The blueprint was created by researching many customer-related scenarios focusing on how to build IT solutions based on the IBM Security Framework. The intent of the blueprint is that it can be used as a road map to assist in designing and deploying security solutions in your own organization.

Building a specific solution requires a specific architecture, design, and implementation; a blueprint can help to create those, *but does not replace them*. Following a blueprint at this point can help identify industry best practices and map them to existing security products and services.

IBM uses a high level service-oriented perspective for the blueprint, based on the IBM Service-Oriented Architecture[2] approach. Services use and refine other services (for example, policy management and access control applies to almost all other services).

Figure 2 illustrates how the IBM Security Blueprint fits within the overall security discussion.

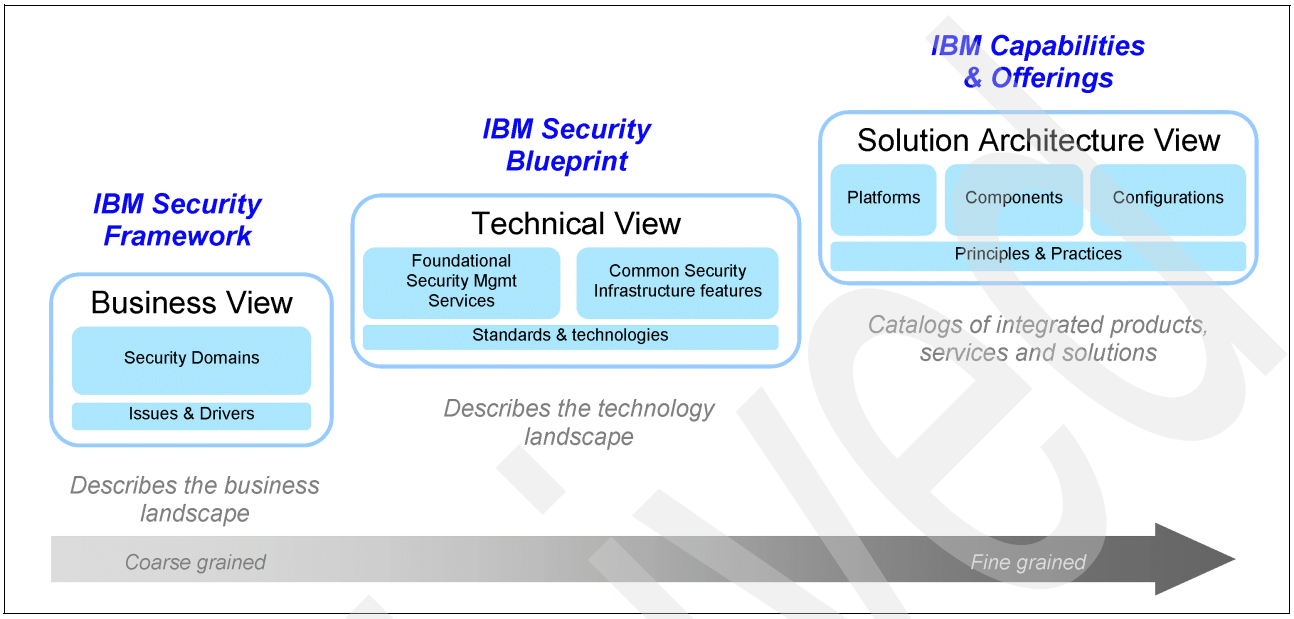

Figure 2 IBM Security Blueprint positioning

The left portion in this diagram represents the IBM Security Framework, which was covered previously, and which defines the security domains. These represent the business view of security concerns.

The middle portion of the figure represents the IBM Security Blueprint, consisting of three building blocks. The *Foundational Security Management Services* describe the top level services needed to achieve the required functionality addressed in the IBM Security Framework. These are the layers where the business requirements, as defined in the framework, are converted to top level IT services to fulfill these requirements. At this point, the threshold has been crossed from a purely business-related viewpoint to actual IT systems.

The *Common Security Infrastructure features* building block contains infrastructure elements and services that are used by the top level services in the Foundational Security Management Services. This building block also encompasses existing infrastructure and systems.

Both of these IBM Security Blueprint building blocks are founded on open *Standards and technologies*.

Both IBM Security Framework and IBM Security Blueprint can help to design the best possible an IT *Solution Architecture View*, which ultimately describes platforms, components, and configurations following architectural principles and practices.

The complete IBM Security Blueprint is shown in Figure 3 on page 6[3].

[2] A comprehensive discussion about the IBM Service-Oriented Architecture can be found in the IBM Redbooks® publication *Understanding SOA Security Design and Implementation*, SG24-7310.

[3] White boxes in this and other diagrams represent services or artifacts that are not solely security related, but may be connected with other IT service areas as well.

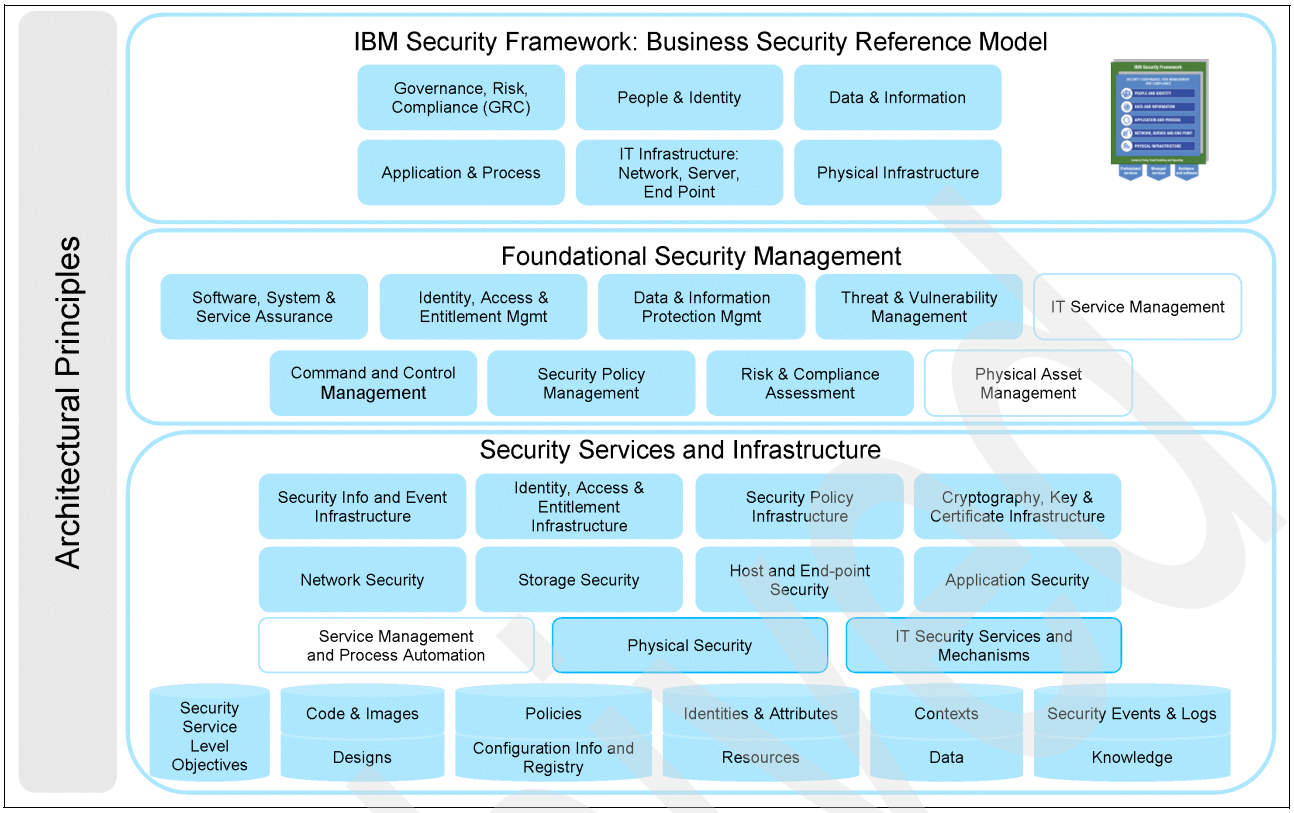

Figure 3 The IBM Security Blueprint

To learn more about the IBM Security Framework and IBM Security Blueprint refer to the IBM Redguide publication *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, REDP-4528.

Data nomenclature

In this section we explain the *data nomenclature* pertinent to our discussion of key life cycle management.

Data at rest

One of the challenges of globalization is the rapidly increasing amount of data that is being generated. It is estimated that storage capacity requirements for large organizations are doubling every 18 months. Data growth comes from many sources: new and growing applications (e-mail, social networking, e-commerce, digital voice, and so on), data generated

and stored due to regulatory compliance requirements (for example, Sarbanes-Oxley[4]), normal organic growth, record retention requirements, and so on. A large organization has to manage all this data wherever it may reside and must focus on:

- **Information availability** - to ensure continuous and reliable access to the data
- **Information security** - to protect and enable the secure sharing of information, to manage the associated business risk, and to ensure personal privacy (84% of security breaches come from within the organization[5])
- **Information retention** - to support the internally and externally required information retention policies
- **Information compliance** - to manage and generate the data to ensure regulatory and audit compliance

An infrastructure of *integrated* IT hardware and software components, along with adequate business *policies* and *processes,* is required to properly address these key focus areas.

In today's environment, this data growth is in part driven by a more complex and collaborative business model, built on a more complex infrastructure (which may include virtualization, SOA, and so on). This means that security also becomes more complex as cyber criminals identify more points of penetration to these global applications. The cyber criminals are also evolving and are no longer the typical fame-hungry hackers operating out of a bedroom in a small town suburb, but organized criminals intent on major fraud and financial gains, and international terrorists or nation states waging electronic warfare against other governments and organizations. The cost of poor security can be quite significant, but the monetary impact may not be the worst of it because the organization also stands to loose brand equity, customer confidence, and intellectual capital, and can face business disruption, the cost of remediation, and the possible costs of legal and regulatory fines.

In summary, a Smarter Planet means a growing and more complex business model and IT infrastructure, which in turn generates more and more data, and this is set to continue for the foreseeable future. The security of this data, wherever it is and however it is kept, must be managed to ensure the survival of the organization. Today, many organizations are turning to encryption as a means of mitigating threats to their data.

Data encryption

All physical storage media (such as disk and tape) will at some point in time be removed from the data center. This may be for maintenance or archiving, to be sent to a business partner, to be returned to the vendor or scrapped at the end-of-life, or it may simply be lost or stolen. As a result, organizations are focussing on using encryption to ensure that the data on such storage media cannot be inadvertently exposed, in particular with storage being built into the infrastructure rather than as an add-on.

There are many ways to encrypt data, but they all basically rely on two components: the *encryption algorithm* (which is usually well documented and well known) and an *encryption key* (see Figure 4 on page 8). The encryption algorithm uses the key to encrypt the data, and also uses the same key (or a mathematically related key) to decrypt the data. For each encryption algorithm there is a very large number of possible keys. For example, when the key length is 128 bits there are up to 2 to the 128th power keys (34×10^{36}). In general it is the key that is kept secret, not the algorithm.

[4] A guide to the Sarbanes-Oxley Act can be found at: http://www.soxlaw.com/

[5] Source: SNIA Data Management Forum, 100 Year Archive Requirements Survey, © Storage Networking Industry Association (SNIA), 2007, http://www.snia.org/forums/dmf/programs/ltacsi/forums/dmf/programs/ltacsi/100_year/100YrATF_Archive-Requirements-Survey_20070619.pdf

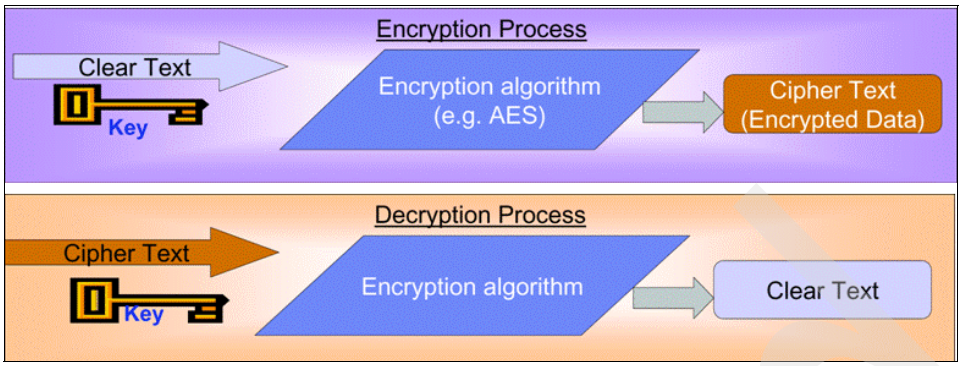

Figure 4 Using encryption to protect data from unauthorized access

We now take a closer look at some of the encryption details:

- Data that is not encrypted is referred to as *clear text*.
- Clear text is encrypted by processing it with a *key* and an encryption algorithm. Several standard algorithms exist, include DES, TDES, and AES.
- Keys are bit streams that vary in length. For example, AES supports 128, 192, and 256 bit key lengths.
- The decryption key can be the same as the encryption key (symmetric encryption) or it can be different (asymmetric encryption).

The key is a critical resource in the encryption process because without it the data cannot be encrypted and, more importantly, it cannot be decrypted. The loss of the decryption key means the data cannot be read and to all intents and purposes it is considered *lost* (or *cryptographically erased*). The way keys are controlled can have a critical impact on the management of storage and information systems. However, storage professionals should not need to become security experts in order to deploy encryption (nor should security professionals need to become storage experts). The storage administrator needs to be assured that implementing encryption will not adversely affect the performance or availability of the storage subsystems (including backup and recovery, disaster recovery, and business continuity processes because this data would also be encrypted).

Encryption management is the management of the *data to be encrypted* and (perhaps more importantly) of the *encryption keys*. Availability of the overall solution is closely tied to the availability of the encryption keys, whereas performance is more tied to the actual encryption process and whether dedicated encryption hardware is used. The focus for key management should be to reduce complexity and cost for encryption.

Key life cycle management

In this section we look at the characteristics of key life cycle management. We describe what criteria constitutes effective key life cycle management and what functions you should look for in a key life cycle management product. Finally, we introduce IBM Tivoli® Key Lifecycle Manager, discuss how it can be implemented in a typical customer IT environment, and provide a technical overview of how it operates.

What is key life cycle management?

Like most IT assets, encryption keys have a life cycle—starting when they are created and ending when they are destroyed.

Before data can be encrypted a key of an appropriate type and length must be created. This key can be used once or many times; it can be used for just one tape or disk, or for many. In order to later decrypt the data, the storage administrator must know which key was used to encrypt that piece of data, while at the same time keeping the key itself secure and confidential (because any exposure of the key would also make the encrypted data accessible and therefore not secure). In addition, there may be government or other regulations specifying the use of keys (such as FIPS 140-2[6] requiring unique keys for each tape cartridge) that must be adhered to.

To increase security you might decide to allocate a key from a pool of keys, allocating keys in sequence based on a round-robin mechanism, and you may wish to change this pool of keys every *n* number of days. To exchange encrypted data with business partners you must be able to exchange keys with them in a secure manner. You must also be able to handle any keys or certificates that have expired (preferably before they expire).

The place where you keep the keys (the *keystore*) must be *highly available* (because without the keystore, your data is inaccessible) and *highly secure* (if the keystore is compromised, then so is the encrypted data), and any disaster recovery site must hold an up to date copy of your keystore, so that you are able to read and create data at your disaster recovery site.

You have to ensure that you do not prematurely delete keys, unless explicit data destruction is required. In fact, you can only safely delete a key when you are sure that all data that it has been used to encrypt is no longer in use. Because it is often very difficult to be certain of this (and the consequence of getting it wrong is that you will loose the data), keys are usually archived rather than destroyed at the end of their lives.

Whatever method is used to manage the keys, it must enable encryption/decryption with minimal performance and administrative overhead.

Managing and keeping track of keys can be challenging. As the amount of data at rest continues to grow, the number of keys also grows. Data-intensive organizations might have many thousands or tens of thousands of keys in use. Managing the life cycle of the encryption keys requires careful thought and planning.

What is needed in a key life cycle management system?

Thinking about the life cycle of an encryption key from the previous section, a key life cycle management system should exhibit the following properties:

- High availability

  The availability of the keystore affects the availability of the data;, its availability must at least match that required from the data.

- High performance

  The first write or read of the data is in fact a write/read to the keystore. Any performance issues with the keystore can affect the overall data access performance, although in general bulk encryption performance is more tied to the implementation and whether there is hardware assistance for the cryptographic processing of the data.

[6] More information about FIPS 140-2 can be found at NIST Web site http://csrc.nist.gov/groups/STM/cmvp/ or in the following PDF file: http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

- Scalability

  Both the number of keys to be managed and the number of devices requesting key services can be expected to increase.
- Security of the keystore

  If the keystore can be compromised, the encrypted data can be as well.
- Secure and reliable key generation and expiration, including:
  - Removal of out-of-use keys
  - Notification about keys that are soon going to expire
  - The ability to respond to a compromised key
  - The ability to securely release keys into production:
    - The ability to make them available to validated systems transparently as needed
    - The ability to restrict access from, and to, all non-validated systems
- Flexibility and efficiency in sending encrypted data to business partners
- Simplicity and transparency
  - The more complex the key management system is, the more likely it is to fail or be implemented badly, either of which increases the risk and cost of encryption.
  - The more simple and transparent a key management system is, the more likely that it will be compatible with existing applications and systems.
- Cost effectiveness
- Provision of full reporting for auditing and management reporting purposes.

In summary, effective key management must accommodate a variety of risks, use cases, application security requirements, and related IT management systems.

The IBM solution to key life cycle management

The *Data and Information* security domain introduced within the IBM Security Framework is our area of interest in this guide. It can be decomposed into the IBM Security Blueprint services and artifacts highlighted in Figure 5, specifically the *Foundational Security Management* disciplines of *Data and Information Protection Management* and *Security Policy Management*.

Within the *Data and Information Protection Management* discipline, one of the underlying *Security Services and Infrastructure* components is the *Cryptography, Key and Certificate Infrastructure*. The corresponding IBM product best suited to implement this service is the *IBM Tivoli Key Lifecycle Manager*.

As shown in Figure 5, the Foundational Security Management layer contains the top level artifacts, which can be directly mapped to the IBM Security Framework. The sublayers themselves consist of multiple individual and linked services that we describe in more detail in the following soon to be published IBM Redbooks publications[7]:

- *IBM Enterprise Security Architecture for People and Identity*, SG24-7751
- *IBM Enterprise Security Architecture for Governance, Risk and Compliance*, SG24-7750
- *IBM Enterprise Security Architecture for Data and Information*, SG24-7752

[7] These IBM Redbooks publications are currently in development and will be published at a later time in 2009.

The components and services highlighted in red, namely *Cryptography, Key & Certificate Infrastructure*, *Data & Information Protection Management*, and *Data & Information*, are the focus of this discussion.

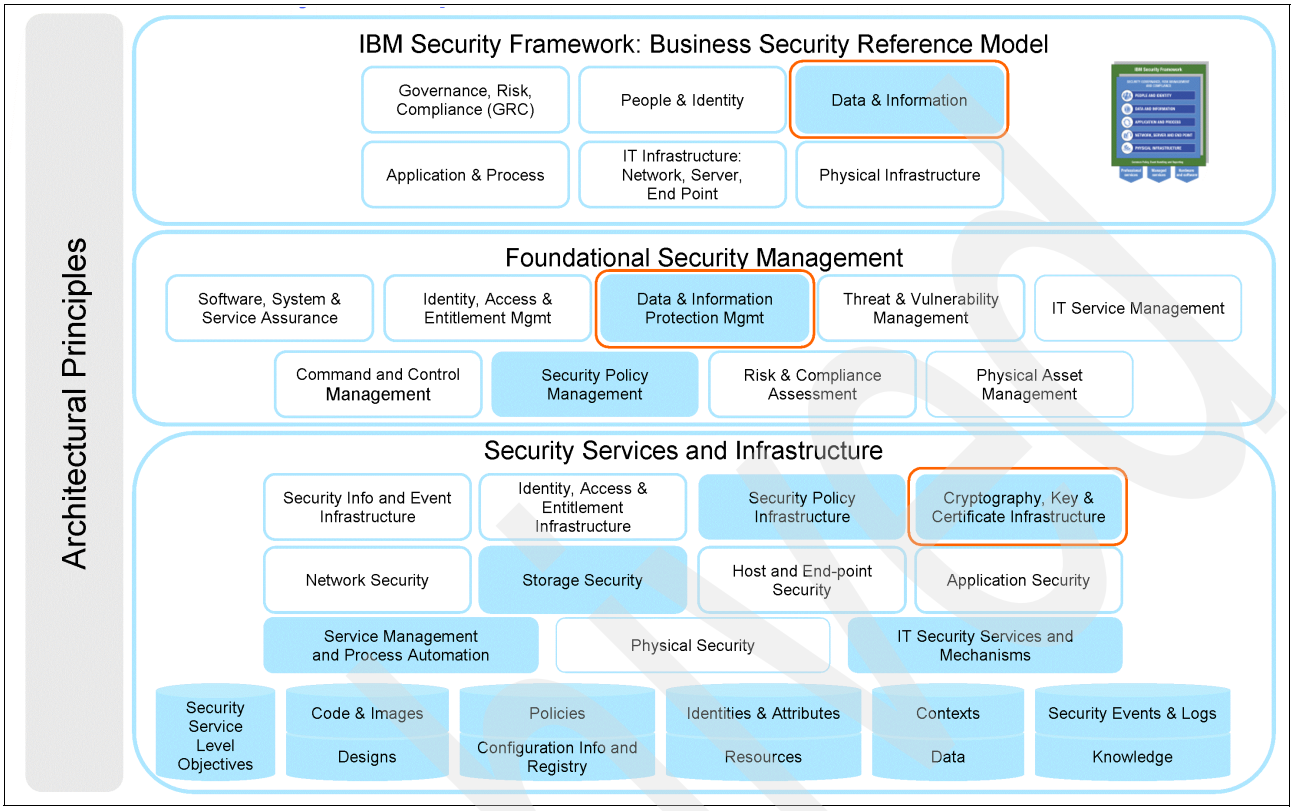

Figure 5 IBM Security Blueprint - Data and Information

One approach to encryption is *application level encryption*, where not necessarily all the data is encrypted, but only a subset that represents the confidential or high-value data. This encryption is often performed by an application itself, by middleware located on one of the servers, or by a specialized appliance. Application level encryption allows you to encrypt at the field level, but requires extensive data classification to determine exactly what needs encrypting. This approach can often lead to significant performance overhead, especially if the encryption is done on the application server itself. In addition, the data classification needs to be re-assessed frequently, as applications and data change, to ensure that the right fields are being encrypted.

IBM found that today 90% of disks that are returned for repair or reuse still contain readable data, and approximately 50,000 disks are currently being retired from data centers every day. This shows why disk-level encryption can be an important consideration when retiring disk storage devices.

Disk or tape level encryption reduces the need for data classification, and with the encryption engine as part of the physical disk or tape drive (not consuming any CPU cycles) the performance impact of encryption can be virtually eliminated. This approach also scales well because each additional disk or tape drive comes with it its own encryption engine. The IBM Tivoli Key Lifecycle Manager has initially focussed on supporting tape and disk level encryption, but it is IBM's stated intention to provide centralized key life cycle management across a range of deployment scenarios.

While many of the current approaches to encryption key management add unnecessary complexity and cost by using a number of different elements to secure the data, the IBM Tivoli Key Lifecycle Manager is a software solution that can run on many industry standard platforms (AIX®, Linux®, Solaris, Windows®, z/OS®). Key management is removed from the data read/write path and thus has no impact on the performance of the data transfer. The solution architecture is straightforward: when a disk or tape drive requires a key (for read or write) it communicates with the IBM Tivoli Key Lifecycle Manager over the network using TCP/IP and requests the required key. IBM Tivoli Key Lifecycle Manager then sends the key to the device over a secure (SSL) session and the data transfer can then proceed, with the encryption itself being performed by the dedicated encryption engine within the storage device.

Figure 6 depicts the logical components of the IBM Tivoli Key Lifecycle Manager.

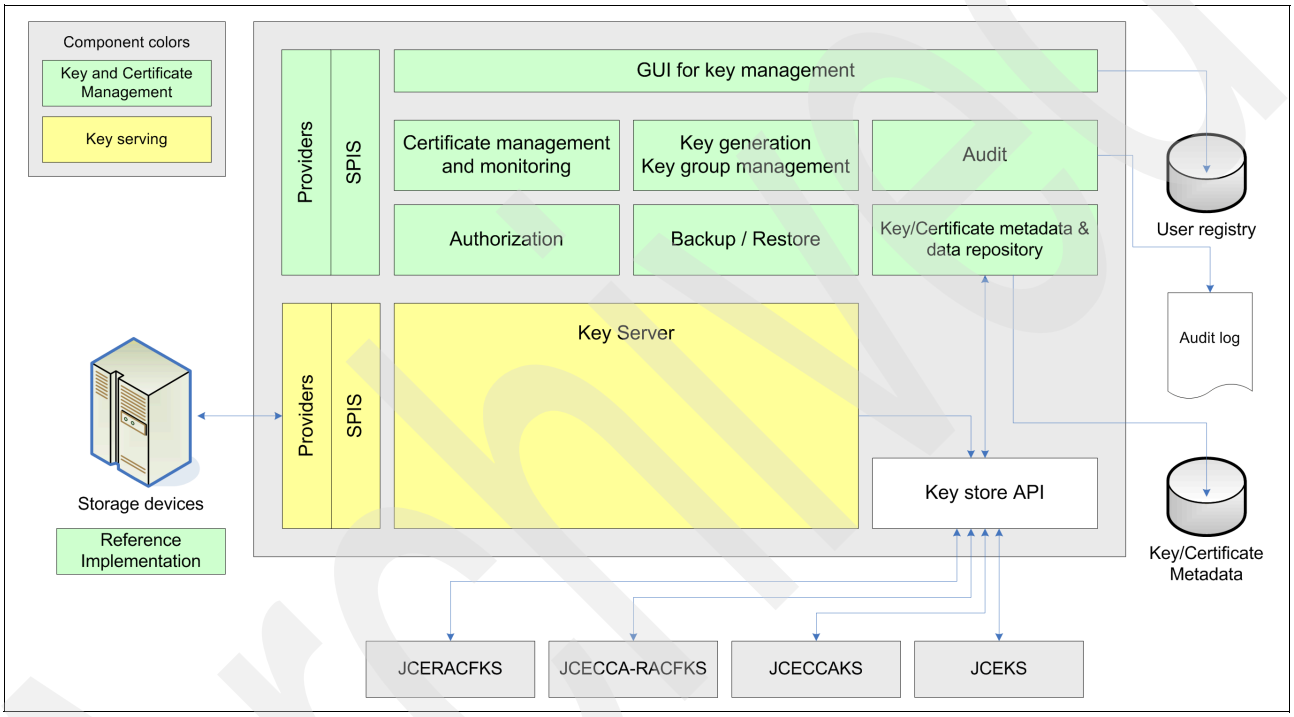

Figure 6 Tivoli Key Lifecycle Manager component overview

IBM Tivoli Key Lifecycle Manager is divided into two major components: *Key Serving* and *Key and Certificate Management*.

The *GUI for key management* is the primary user interface into IBM Tivoli Key Lifecycle Manager and provides all the functionality for day to day key management. The *Audit* component creates and sends audit information into an external *Audit Log* and the *Backup/Restore* facility enables backup of the IBM Tivoli Key Lifecycle Manager keystore and associated metadata. The *Key/Certificate metadata and data repository* function maintains the metadata associated with the keys (such as key identifier, the tape serial number, and so on) and stores it in a DB2® database. Keys are generated and key grouping maintained by the *Key generation, Key group managemen*t component, which also manages key group rotation. *Certificate management and monitoring* handles certificate management, for example, the creation of certificates or the obtaining of them from third parties, checking the validity of certificates (that they have a valid root certificate and have not expired) and manages their renewal.

IBM Tivoli Key Lifecycle Manager can be implemented to only serve keys to predefined storage devices, and the *Authorization* function to validate these devices as necessary. It is IBM's stated intention to provide an API (the *Providers SPIS*) to enable communication with other key managers.

The Key Serving component is responsible for obtaining the correct key from the keystore using the *Key Store API* and delivering it securely to the encrypting storage device. In addition, IBM is helping to lead a standards effort called the *Key Management Interoperability Protocol,* being worked on in the Oasis standards group, which in the future might enable a wide variety of integration possibilities.

A typical Tivoli Key Lifecycle Manager implementation

In this section we take a closer look at the conceptual Tivoli Key Lifecycle Manager implementation. For some real-world deployment scenarios refer to "Customer scenarios" on page 17.

Tivoli Key Lifecycle Manager can run on a standalone server independent of any application server as shown in Figure 7. It can be implemented on a number of different operating system platforms. Check the product's online Information Center[8] under **Product Overview** → **Release Information** for specific details.

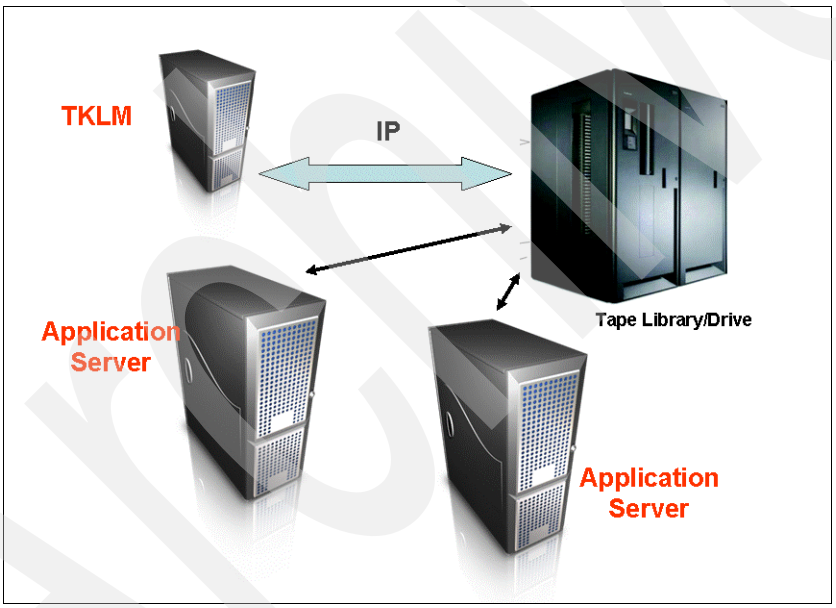

Figure 7 Centralized encryption key manager

When an application needs to write to a tape, as illustrated in Figure 7, it issues a *tape mount request* to the tape library. The tape library communicates with Tivoli Key Lifecycle Manager to retrieve the applicable encryption key. When the tape library has received the key and the tape cartridge is mounted, the tape library signals the application that the tape is now available. The application starts sending the data to the tape library and the data is encrypted by the drive using the obtained key. The application has no knowledge that the data is being encrypted and therefore no changes are required to the application. In this way the use of Tivoli Key Lifecycle Manager is transparent to the application.

Figure 8 provides more detail on how the key is securely sent to the tape library.

[8] http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tklm.doc/welcome.htm

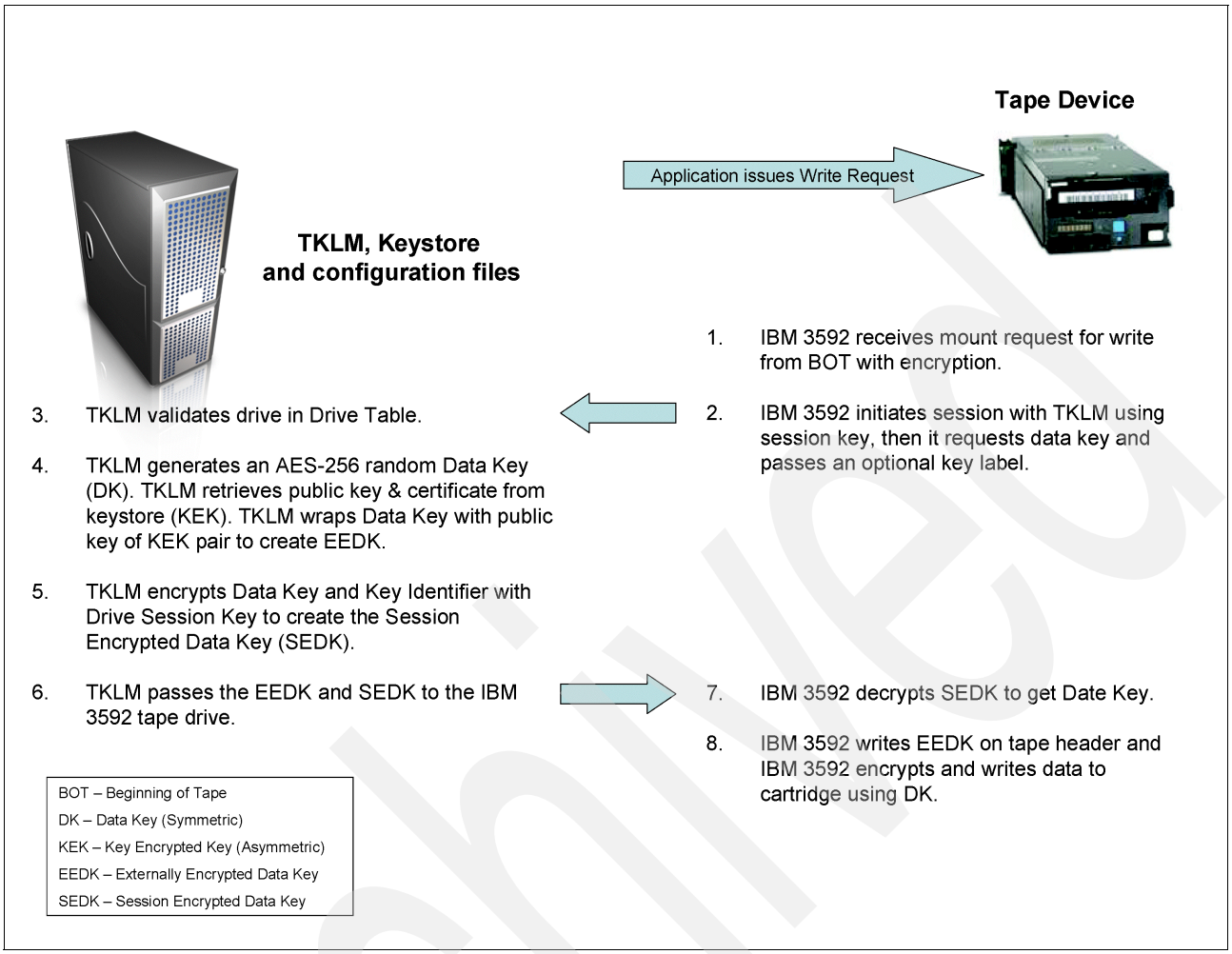

Figure 8 IBM 3592 tape write encryption process

The *data key* (a symmetric key) is sent to the drive encrypted by the drive's public key. The drive decrypts the SEDK using the session key to obtain the data key, which it then uses to encrypt the data. Tivoli Key Lifecycle Manager also sends the data key, encrypted with one of Tivoli Key Lifecycle Manager's public keys (the *EEDK*). This EEDK is written to the tape and can later only be decrypted by the holder of the corresponding private key.

The tape now contains the application data (encrypted by the symmetric data key) and a pair of EEDKs. This way the data key only exists in the EEDK on the tape after it has been written to the tape; it is not stored in the Tivoli Key Lifecycle Manager keystore or on the tape device.

Subsequently, decryption is required when an application wants to read the data or append to an existing tape. Figure 9 on page 15 depicts how this is achieved, again, transparently for the application.

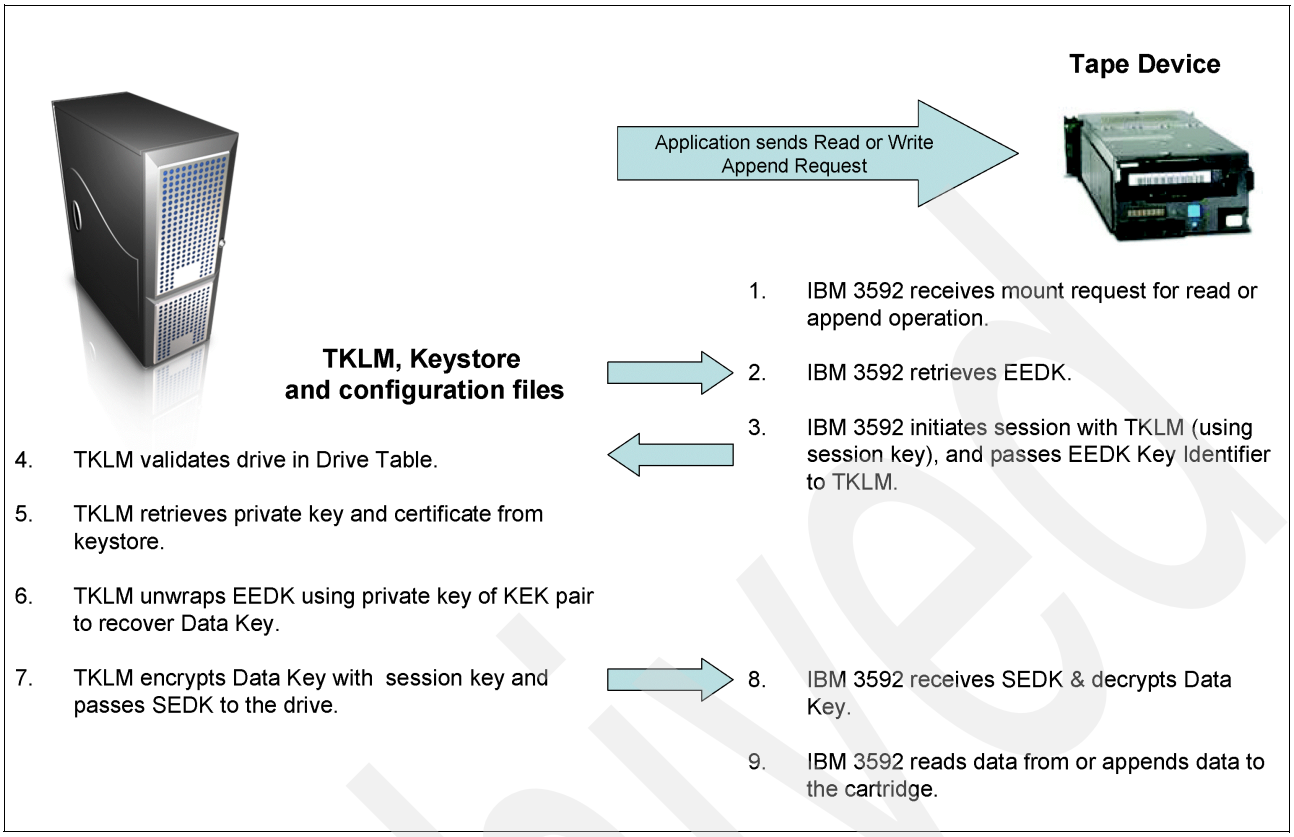

Figure 9 IBM 3592 tape read/append decryption process

The EEDK is retrieved from the tape and sent to Tivoli Key Lifecycle Manager, which owns the corresponding private key. Tivoli Key Lifecycle Manager decrypts the EEDK using this private key to retrieve the Data Key, which it then sends to the tape drive securely encrypted by the drive's public key (in an SEDK).

To send an encrypted tape to a business partner, all that is required is the public key from the business partner to create the EEDK, rather than the public key from your own Tivoli Key Lifecycle Manager. In fact, Tivoli Key Lifecycle Manager allows you to specify two keys for the drive, your own public key and your business partner's public key, thus two EEDKs are created. This way both you and your business partner (but *only* you and your business partner) can decrypt the tape.

The process is similar and differs only slightly for other device types, such as LTO-4 tapes or IBM DS8000® disks.

We now take a closer look at how Tivoli Key Lifecycle Manager meets some of the requirements for a key management solution as identified in "What is needed in a key life cycle management system?" on page 9.

- High availability

  Disk drives only need to communicate with Tivoli Key Lifecycle Manager once at power up when they obtain the key. Most of the time these devices can operate independently from Tivoli Key Lifecycle Manager. Also, multiple Tivoli Key Lifecycle Managers can be configured (primary and secondaries). If one of these systems becomes unavailable, there are one or more secondary systems available to take over the key serving task. In addition, because Tivoli Key Lifecycle Manager is network attached to the device, the secondary Tivoli Key Lifecycle Managers can reside in remote locations if needed. Tivoli

Key Lifecycle Manager provides management tools to enable the primary and secondary systems to be kept synchronized. Tivoli Key Lifecycle Manager also provides management tools to allow full backup and restore of all its critical data.

- High performance

  Tivoli Key Lifecycle Manager is typically deployed on a dedicated server. The connectivity to the encrypting device is usually achieved over a fast network, and the data volumes (used to deliver the keys to the storage devices) are relatively small and independent of the amount of data being encrypted, so network overhead is negligible.

- Scalability

  By far the main potential overhead in an encrypting solution is the actual task of encrypting the data, which is performed by dedicated encryption hardware in each storage device (and not by Tivoli Key Lifecycle Manager). More supported storage devices can be added to this solution in the future, and each one can bring its own encryption engine. This leads to a very scalable solution.

- A secure keystore

  All versions of Tivoli Key Lifecycle Manager support the userid/password protected JCEKS keystore in addition to any platform-specific security. In the z/OS environment, three additional keystore types are supported.

  One of these is the JCERACFKS keystore, which makes use of all the security advantages of RACF® by storing keys and certificates in a RACF database. Use this keystore type to store key material in your RACF keyring that is not using the Integrated Cryptographic Services Facility (ICSF). Because this keystore does not support symmetric keys, it cannot be used with LTO drives. Selecting the appropriate keystore for your environment is driven by the regulations and requirements that your business must meet.

- Secure and reliable key generation

  Tivoli Key Lifecycle Manager provides a Web application based graphical user interface (GUI) for the creation and management of keys. It provides warnings when certificates are due to expire so they can be renewed or replaced in good time. Tivoli Key Lifecycle Manager also validates the devices to which it provisions keys, transparently delivering keys to those which it is authorized to work with while denying key serving to all others.

  For increased security, Tivoli Key Lifecycle Manager also enables the periodic rotation of keys or groups of keys. For example, a tape drive can use keys in a round-robin fashion from one group for a month and then from another group the following month, ensuring that the organization data is encrypted using many different keys. The Tivoli Key Lifecycle Manager administrator can set up this rotation schedule many months in advance to reduce the amount of day-to-day administration tasks.

  To provide flexibility and efficiency in sending encrypted data to business partners, Tivoli Key Lifecycle Manager enables the safe and secure transport of encrypted tapes between business partners by, for example, supporting the use of two key encrypting keys (described with Figure 9 on page 15). The first one is Tivoli Key Lifecycle Manager's own public keys, the second is the public key of the business partner. The tape can then be sent to the business partner encrypted, and decrypted *only* by the holder of the private keys, namely the business partner or the sending organization.

- Simplicity and transparency

  Tivoli Key Lifecycle Manager runs on a server that is effectively *out-of-band* to the data path. The applications have no awareness of the data being encrypted. They send the data to the disk, and the disk then handles the communication with Tivoli Key Lifecycle Manager to encrypt the data. The whole encryption process is transparent to the application.

In addition, the Tivoli Key Lifecycle Manager environment is easy to install and use, with an installation wizard reducing installation time to less than an hour for the distributed version of Tivoli Key Lifecycle Manager, and it is administered and controlled by a Web application based GUI.

- Cost effective

  Tivoli Key Lifecycle Manager allows key management to be accomplished in a very efficient way and with little on-going administration required. It can largely be considered *install and forget*.

- Audit reporting

  In order to provide proof of encryption full audit logging is available. An audit record can be created for every encryption event.

Customer scenarios

In this section we describe two real-world customer scenarios. The first is a straightforward implementation of Tivoli Key Lifecycle Manager to manage the keys for encrypting tape drives. The second scenario is a more demanding implementation of Tivoli Key Lifecycle Manager on z/OS to manage the keys for encrypting disk drives.

Customer One

Due to regulatory compliance requirements Customer One is challenged to encrypt all their customer-related personally identifiable information when it gets stored on local tape drives. In addition, Customer One also sends archive tapes off site for disaster protection. They need to ensure that these tapes cannot be accessed by anyone other than the originating data center.

A secondary requirement for Customer One is the availability of the solution. Due to their 24/7 worldwide online business activities, the customer data applications must be able to access the encrypted data on their tape drives at all times.

As a non-functional requirement Customer One wants to securely locate the encryption key management systems within their secure management network zone. In the past Customer One had experienced a disgruntled employee penetrating some IT systems located in their production network zone.

Solution architecture

Customer One's applications are deployed on an AIX system, and it is to this machine that they attach the IBM 3592 tape drive library. This system is located in the production network zone, and external as well as internal access to the Web-based applications is channeled through a Web security server function[9] deployed in the Internet DMZ (for external access) and the production zone (for internal employees). The application server accesses the tape drives without the need to know that Tivoli Key Lifecycle Manager will be involved to provide encryption keys.

In the first phase one Tivoli Key Lifecycle Manager server is being deployed within the management zone on an AIX system (due to familiarity with that platform). The communication between the tape library and Tivoli Key Lifecycle Manager is implemented

[9] The Web security server can be an IBM Tivoli Access Manager for e-business WebSEAL or Web server plug-in component. The infrastructure layout for this component is omitted from this discussion. For more information refer to the IBM Redbooks publication *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014.

using a secure SSL connection with a custom port. In addition, the firewall between the production and management zones (which is not explicitly depicted in the architecture diagram) is configured to only allow traffic for this custom SSL port coming from the IP location of the tape library, effectively limiting application access to the Tivoli Key Lifecycle Manager system, and by that, securing the encryption keys stored on the Tivoli Key Lifecycle Manager system.

In this first phase the administrator's access to the Tivoli Key Lifecycle Manager configuration will only be allowed through an administrative workstation that is located in the management zone. No access from outside that zone will be allowed, which further strengthens the security by isolating the encryption keys.

Administrative option: Customer One is also investigating routing of the administrative access to Tivoli Key Lifecycle Manager through the regular internal Web security server to avoid requiring the administrator to go to a specific physical location for his administrative duties. This can be achieved, for example, by creating an Access Manager WebSEAL junction to the Tivoli Key Lifecycle Manager administrative application that runs on the embedded WebSphere® Application Server on the Tivoli Key Lifecycle Manager system.

To strengthen the access control for administrative access to the Tivoli Key Lifecycle Manager administrative application Customer One can configure a step-up authentication mechanism via WebSEAL by using, for example, biometric access control.

After the first phase has been tested, a second Tivoli Key Lifecycle Manager server is added to allow for high availability in case one of the systems becomes defective or unresponsive. The Tivoli Key Lifecycle Manager key export/import function is used to synchronize the two keystores once the second system has been installed. After the libraries have been configured with a list of Tivoli Key Lifecycle Manager TCP/IP addresses, automatic switching can occur if there is a failure.

The overall architecture diagram is presented in Figure 10.

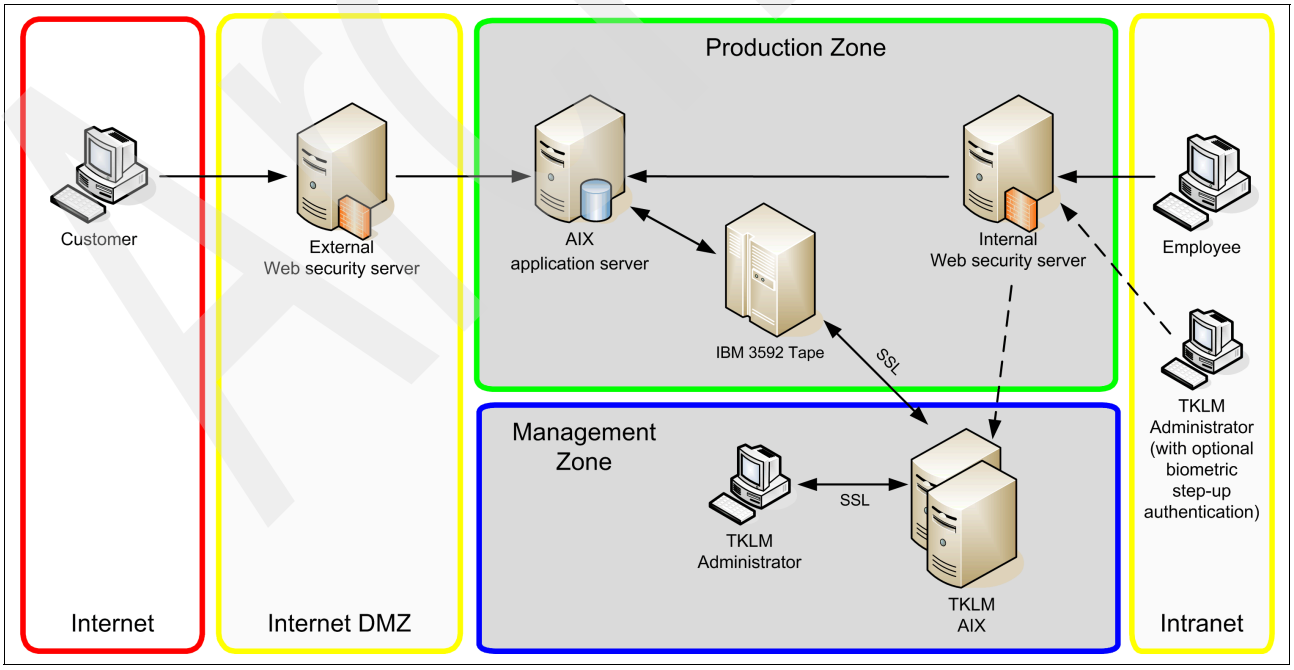

Figure 10 Customer One solution architecture

Customer One can be considered a small/medium organization with a very small IT support team and no prior experience using an encryption key manager. However, this deployment architecture enabled them to successfully install Tivoli Key Lifecycle Manager in their environment with minimal IBM assistance. Once IBM Tivoli Key Lifecycle Manager is set up very little on-going administration is required.

Customer Two

Customer Two is using an IBM System z® server to host their business applications running on z/OS. They are already using an existing key manager (IBM Encryption Key Manager) to encrypt data that is stored on tape units. Customer Two is planning to migrate their disk drives to IBM DS8000 encrypting disk drives for the extra capacity, improved performance, and the encryption capability that these drives deliver. This way they will have an encompassing encrypted data strategy from disk to tape, which has been mandated by their senior management due to the sensitivity of data they are dealing with. Their existing IBM Encryption Key Manager does not support the IBM DS8000 disk drives, so they need to migrate to IBM Tivoli Key Lifecycle Manager.

The deployment plan is split into two distinct phases: a test phase and the production deployment. First Customer One wants to install the DS8000 in a test environment, seed it with test data, and evaluate disk performance while using Tivoli Key Lifecycle Manager for z/OS to manage the keys. The first proposed configuration is depicted in Figure 11.

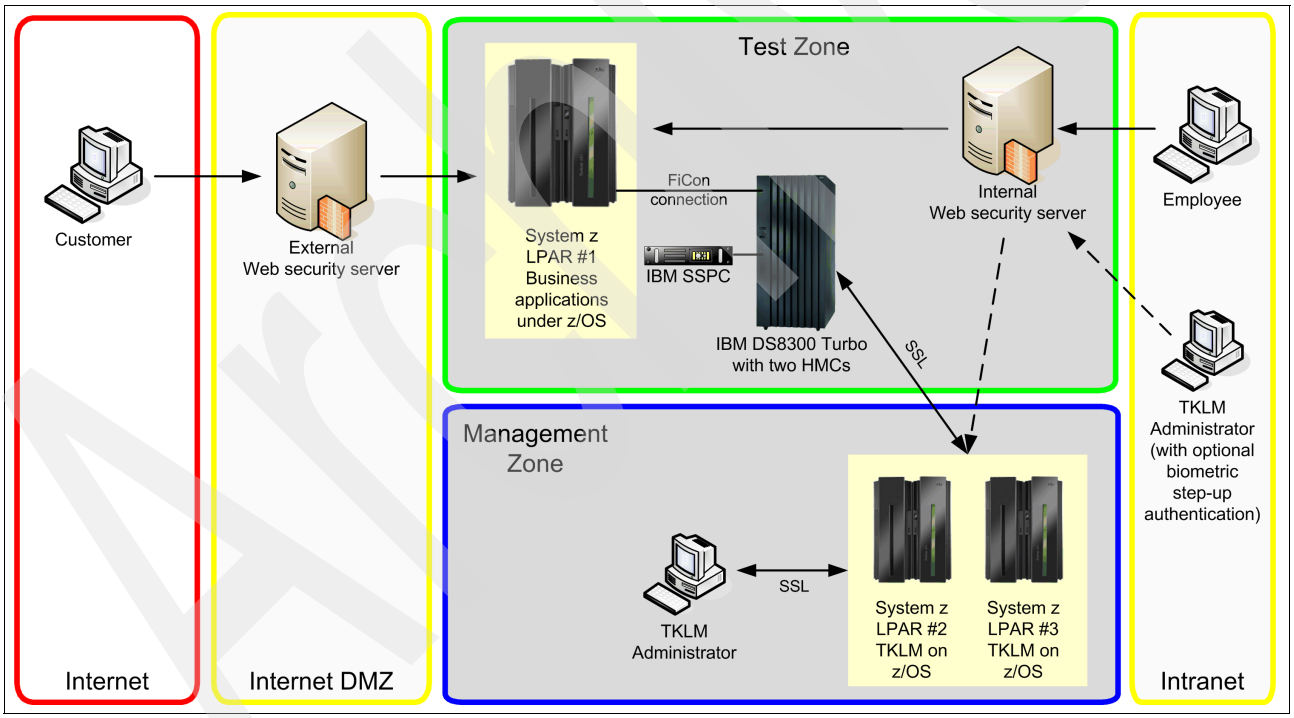

Figure 11 Customer Two proposed initial architecture

The DS8000 disk drives require that at least two Tivoli Key Lifecycle Managers are defined for availability reasons (the DS8000s will not allow you to configure only one Tivoli Key Lifecycle Manager). The depicted IBM SSPC server is needed for DS8000 management purposes only and has no impact on the key management function.

The pictured initial configuration has two Tivoli Key Lifecycle Managers, each running on a separate LPAR on the System z. In the architecture diagram they are shown as being configured in a network zone different than the System z application server LPAR (due to best

practices to keep encryption keys in a more highly secured network zone); however, they are still being implemented within the same physical system.

Since Customer Two is running a highly virtualized IT environment where the storage is dynamically managed, we are facing a potential complication in this architecture. It is very possible that the Tivoli Key Lifecycle Manager or z/OS system datasets could be automatically (and accidentally) migrated onto the encrypted DS8000 disk drives. This can lead to a *lock-out* situation on power up in the following two scenarios:

1. If the z/OS system datasets are encrypted:

   To IPL z/OS, its system datasets on the DS8000 must be accessed. Because these are encrypted, the DS8000 tries to access Tivoli Key Lifecycle Manager, but Tivoli Key Lifecycle Manager cannot start until z/OS is started (and z/OS cannot start until Tivoli Key Lifecycle Manager is started to retrieve the encryption keys)—hence, a lockout.

2. If the Tivoli Key Lifecycle Manager datasets are encrypted:

   In this case, z/OS can IPL, but for Tivoli Key Lifecycle Manager to start it needs access to its datasets on the DS8000, which it cannot get to until Tivoli Key Lifecycle Manager is started to retrieve the encryption keys—hence, a lockout.

To address this situation, encrypting DS8000s are delivered with a stand-alone Tivoli Key Lifecycle Manager on an IBM System x® server running Linux. The test configuration architecture for Customer Two is depicted in Figure 12.

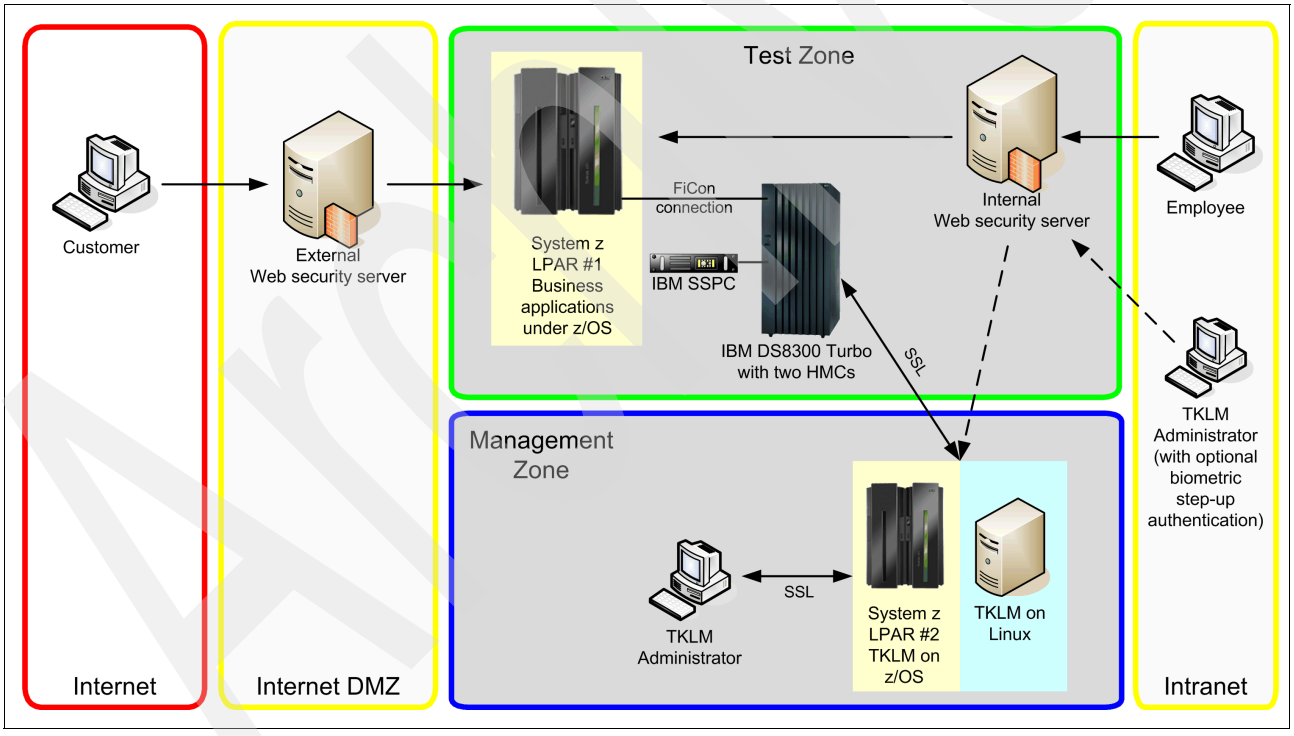

Figure 12 Customer Two phase-1 test

In the DS8000 configuration the Tivoli Key Lifecycle Manager on z/OS is defined as the primary key manager and the one on the Linux system as the secondary. The encryption requirement are:

- Performance
  The use of encryption technology should not impact application performance.
- Availability
  The use of encryption technology should not impact availability.

When Customer Two investigated the performance levels in the test environment they found that despite the full disk being encrypted, there was no significant degradation in application responsiveness. Encryption itself may add 1-2% to disk response times (this factor is so small because encryption is performed at the disk drive using specialized encryption hardware) but the improved base performance of the DS8000 in comparison to their previous disk drives more than made up for this.

Next they looked at the availability of the configuration; the new unknown factor was Tivoli Key Lifecycle Manager. They tested its availability characteristics by simulating various failure scenarios (such as loss of one Tivoli Key Lifecycle Manager before power up, loss of a Tivoli Key Lifecycle Manager when in operation, the ability to move keys between the Tivoli Key Lifecycle Managers, the loss of network connection, and so on) and all these tests were successful—meaning that the DS8000 was able to continue working in all cases. Customer Two also found that, once configured and tested, the IBM Tivoli Key Lifecycle Manager required very little ongoing administrative intervention.

After testing was completed there were two choices for the migration to a live environment:

- Remove all data from the DS8000, remove the Tivoli Key Lifecycle Manager, and start again in the production environment by installing a new Tivoli Key Lifecycle Manager on z/OS.
- Keep the data on the DS8000, back up the Tivoli Key Lifecycle Manager data (using the built-in backup facility), and restore it to a Tivoli Key Lifecycle Manager on a production LPAR.

In this scenario Customer Two decided to follow the first route—basically, to start again with a new and pristine system.

Once deployed in the production environment, Customer Two also wanted to incorporate a remote disaster recovery site, to provide additional Tivoli Key Lifecycle Manager backup and availability. Because Tivoli Key Lifecycle Manager can be considered a completely separate entity, running on dedicated servers and communicating with the storage media *out-of-band*, this could be achieved with all four Tivoli Key Lifecycle Managers defined in the DS8000 configuration as shown in Figure 13 on page 22.

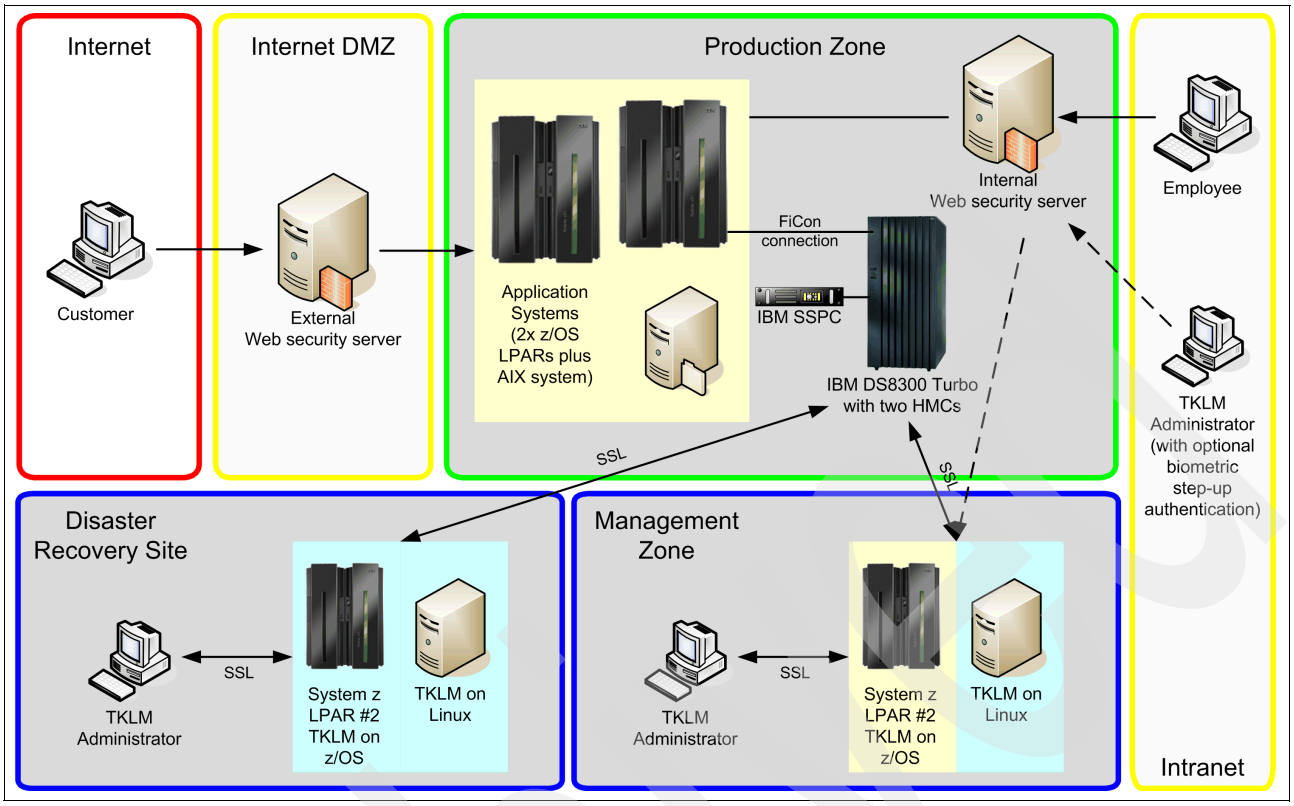

Figure 13 Customer Two production deployment architecture

There was one final step to take. The tape units were still using IBM Encryption Key Manager as their key manager. To migrate them to use Tivoli Key Lifecycle Manager's control required that the tape drives be re-configured to point to Tivoli Key Lifecycle Manager as their key manager, the tape drives be defined to Tivoli Key Lifecycle Manager, and the keys migrated (export/import) from IBM Encryption Key Manager to IBM Tivoli Key Lifecycle Manager. The tape drives have been omitted in our architecture diagrams.

Conclusion

While the amount of data at rest continues to grow, so too will the demand for data encryption. The encryption keys are an integral and vital part of this strategy and need to be carefully managed, otherwise data loss or data exposure can occur. A good key management solution can provide whole life cycle management in an effective, centralized, and transparent way.

IBM Tivoli Key Lifecycle Manager is a full life cycle encryption key manager that addresses many of the requirements and concerns of storage, security, and data governance teams:

- It can reduce encryption management costs related to set up, use, and expiration of keys.
- It can enable organizations to comply with disclosure laws and regulations.
- It can prevent loss of information due to key mismanagement.
- It can transparently detect encryption-capable media to assign necessary authorization keys.
- It can run on most existing server platforms to leverage the server's existing access control/high availability/disaster recovery configurations.

Where to go next

Contact your IBM representative for more information about how the IBM Tivoli Key Lifecycle Manager can benefit your organization.

To learn more about the IBM Security Framework and IBM Security Blueprint refer to the IBM Redguide publication *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, REDP-4528.

More detail about the decomposition of the IBM Security Blueprint as well as product details for other IBM security products can be found in the following IBM Redbooks publications[10]:

- *IBM Enterprise Security Architecture for People and Identity*, SG24-7751
- *IBM Enterprise Security Architecture for Governance, Risk and Compliance*, SG24-7750
- *IBM Enterprise Security Architecture for Data and Information*, SG24-7752

Security solution overviews provided by IBM can be found in the following IBM Redbooks publications:

- *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014
- *Enterprise Security Architecture Using IBM ISS Security Solutions,* SG24-7581

To find more information about IBM Tivoli Key Lifecycle Manager please visit the following resources:

- IBM Redpaper™ *IBM Tivoli Key Lifecycle Manager for z/OS*, REDP-4472

  http://www.redbooks.ibm.com/redpieces/abstracts/REDP4472.html?Open
- Product overview

  http://www.ibm.com/software/tivoli/products/key-lifecycle-mgr/
- Product documentation

  http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tklm.doc/welcome.htm

The team who wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Axel Buecker is a Certified Consulting Software IT Specialist at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on areas of Software Security Architecture and Network Computing Technologies. He holds a degree in Computer Science from the University of Bremen, Germany. He has 23 years of experience in a variety of areas related to Workstation and Systems Management, Network Computing, and e-business Solutions. Before joining the ITSO in March 2000, Axel worked for IBM in Germany as a Senior IT Specialist in Software Security Architecture.

David Crowther has 30 years of experience in the IT industry, the last 23 working for IBM. During his IBM career, he has worked in technical pre-sales, services, and support, and currently works in IBM BetaWorks, where he manages early beta programs for Tivoli Security and Provisioning products. In addition, he runs enablement workshops, provides technical support, and acts as a subject matter expert for new products. He also has wide experience in running beta programs on and supporting products from many of the other IBM brands,

[10] These IBM Redbooks publications are currently in development and will be published at a later time in 2009.

including large systems, networking, pervasive, Lotus®, voice, and WebSphere. He is a Consulting IT Specialist, Chartered IT Professional and Chartered Engineer, and holds a Master's degree in Electrical Sciences from Cambridge University.

Thanks to the following people for their contributions to this project:

Alison Chandler
International Technical Support Organization, Poughkeepsie Center

Gordon Arnold, Krishna Yellepaddy
IBM

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document, REDP-4529-00, was created or updated on October 27, 2009.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at
http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
DB2®
DS8000®
IBM®
Lotus®
RACF®
Redbooks®
Redguide™
Redpaper™
Redbooks (logo) ®
Smarter Planet™
System x®
System z®
Tivoli®
WebSphere®
z/OS®

The following terms are trademarks of other companies:

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.