Key Management in the Multi-Platform Environment

A Practical Guide to Choosing a Key Management Solution for Your Enterprise

**THIS WHITE PAPER** discusses the challenges of deploying key management for encryption in business applications. The modern enterprise deploys a variety of server platforms, operating systems, and programming languages. A major barrier to deploying encryption has been the challenge of accessing encryption keys from these widely divergent environments. This paper will help define the key management needs of your enterprise, while defining the elements necessary to reduce the technical challenges and cost of deployment.
**Key Managers and Key Retrieval**

Encryption key management solutions have the primary goal of managing and protecting encryption keys, and making them available to authorized applications in a secure fashion. Key retrieval includes the process by which an application requests an encryption key from the key server, how the key is protected during the delivery of the key, and the format of the encryption key and policy information that is delivered.

Key management solutions vary greatly in the complexity of the key retrieval process. The more complex the key retrieval interface, the greater the challenge for the enterprise IT team in deploying key retrieval in applications. Understanding this fact can help IT decision makers assess different vendor solutions and the likely costs of deploying a solution in their enterprise.

**KMIP Standard Key Management**

Industry standards are important for customers who want to achieve good security and a measure of independence from any one key management vendor. The OASIS Key Management Interoperability Protocol (KMIP) is an important key management industry standard that is supported by all professional key management systems.

**Payload Complexity**

When a key server provides an encryption key to a client application, it packages the encryption key with additional information about the key. This additional information may include usage policy (can the key be used for encryption, digital signature, and so forth), key expiration information, and key access information. The packaging of this information may be more complex or less complex. While more complex packaging can provide more flexibility, it can also reduce the ability of developers to use the contents of the package on new platforms. A complex package can represent a “Rubik’s cube” challenge for developers – slowing deployment and delaying projects.

In some cases Enterprise customers have been forced to hire specialized consultants from the key management vendor just to implement the application interface for key retrieval. In this case the consulting costs can quickly exceed the costs of the key management solution itself.

**Sample Source CodeSpeeds Deployment**

One way that vendors can help their customers deploy key management solutions quickly is to provide sample source code and platform-specific programming guides. Sample source code will help a developer quickly understand how to implement the key retrieval solution in your applications, and understand the best practices for programming to the vendor’s interface. Sample source code also provides the developer an easy means of validating the implementation. If the sample code is adequate, it can be compiled and demonstrate successful key retrieval in a very short period of time.

When a vendor provides sample source they are also making a statement about the ease of use of their product. Sample source code provides a road map for the developer and shows the simplicity (or complexity) of the implementation. This gives IT professionals a good yardstick to determine the actual cost of the implementation.

**Key Retrieval From Windows Applications**

The windows platform can represent the biggest challenge from a key retrieval perspective. There are many programming languages on the platform, and many versions of the windows operating system. At a minimum the following language environments should be supported for key retrieval:

- C and C++
- C# (C Sharp)
- Java

This wide array of languages presents a daunting challenge to a key management vendor. A well thought-out strategy is needed to be sure that encryption keys can be retrieved into all of these environments.

Since Microsoft has clearly defined the .NET platform as their strategic development platform, the key management vendor should provide a .NET assembly as a core platform for key retrieval. The windows developer will be able to incorporate the assembly into any of the modern Microsoft languages.

In addition to support for the newer .NET assembly, a key management vendor should also support the traditional
Dynamic Link Library (DLL). DLLs are used in older Microsoft and third party languages. There are many thousands of legacy applications that use this older approach to application development, and support for this approach is crucial for success on the Microsoft Windows platform. The many versions of Microsoft Windows can also present a challenge to the key management vendor. Typically the enterprise customer will run a variety of third party solutions which require different version of the Windows server and PC operating systems. It is not uncommon to see applications running on Windows NT or even earlier versions of the Windows operating system. Minimally, you should see support for the following Microsoft platforms:

- Windows 2003
- Windows 2008
- Windows 2008 R2
- Windows Vista
- Windows 2012
- Windows 2012 R2

Key Retrieval From Microsoft SQL Server

Recognizing the importance of proper key management for data security, Microsoft implemented Extensible Key Management (EKM) in SQL Server 2008. Available in SQL Server 2008-2014 Enterprise Edition and higher, EKM is both an architecture for encryption key management services, as well as an interface for third-party encryption key managers (both Hardware Security Modules (HSMs) and virtual appliances in the cloud). While EKM provides for local, on-server management of encryption keys, Microsoft and third-party security professionals recommend the use of external key management.

Along with EKM in SQL Server 2008/2012/2014, Microsoft provides two options to users for encryption – Transparent Data Encryption (TDE) and Cell Level Encryption. It is essential for any encryption key manager to support both of these encryption options.

For users of non-EKM versions of SQL Server, your key management vendor should be able to support and provide sample code your programming environment.

Key Retrieval From Oracle Database 10g/11g

Oracle’s MySQL database is a popular open source database in use in a large number of Linux and Windows environments. MySQL customers use a variety of development languages for accessing MySQL data including Java, .NET languages, Perl, PHP, Ruby, and others. MySQL customers can use the native development language support for encryption, and use a vendor’s key management solution for key management and key retrieval. Your key management vendor should be able to support your development language environment and provide you with sample application code.

Key Retrieval From Linux Applications

There are a variety of languages used for applications on the Linux platform. Java, C and C++, Perl, and PHP are some of the popular languages. Java is a special case and is covered in more detail below. In spite of the number of languages available on Linux, the job of providing key retrieval support is a bit easier for the key management vendor. Linux programming languages support shared libraries, and the key management vendor should provide this type of support for key retrieval.

Key Retrieval From IBM i DB2

The IBM i (AS/400, iSeries, System i) server platform is found in most enterprise environments providing support for back office applications. The programming languages used on this platform include RPG and COBOL, with the large majority of applications built using the RPG language. Encryption key retrieval should support both of these languages.
An effective way to provide key retrieval support on the IBM i platform is by using service programs. Service programs are very similar to shared libraries and Dynamic Link Libraries (DLLs). They can be easily added to most RPG and COBOL applications. The key management vendor should provide key retrieval support using a service program so that the IBM i developer can use it with any IBM i application programming language.

In the latest versions of the IBM i operating system (V7R1 and V7R2) IBM provides a DB2 automatic encryption facility named FIELDPROC (Field Procedures). This facility allows IBM i customers and vendors to provide encryption and key retrieval software to automatically encrypt one or more columns in the DB2 database without application changes. Your IBM i software vendor should provide you with the encryption software and the key manager needed to protect data.

Key Retrieval From IBM DB2

For non-IBM i platforms IBM’s DB2 database is available for Windows and Linux platforms. IBM customers use a variety of development languages for accessing DB2 data including Java, .NET languages, and others. DB2 customers can use the native development language support for encryption, and use a vendor’s key management solution for key management and key retrieval. Your key management vendor should be able to support your development language environment and provide you with sample application code help you get started.

Key Retrieval From IBM z

The IBM z (Mainframe, z/OS, OS/390) platform runs a variety of operating systems and supports a variety of programming languages. Most enterprise customers run the z/OS operating system and use Cobol for their back office applications. Key retrieval support for this environment is crucial for the enterprise customer. However, there are often applications written in other languages such as PL/I and Assembler.

The key management vendor should provide support for key retrieval both as a Dynamic Link Library and as an object module.

It is especially helpful to the mainframe developer to have sample Cobol source code as a guide for implementing key retrieval via a DLL interface. The sample code will help the developer understand the DLL architecture for key retrieval, and provide a quick method to validate key retrieval in the more complex z/OS environment.

Key Retrieval From Java

The Java language is popular because applications written in this language can be deployed on a wide variety of hardware platforms and operating systems. Key management vendors have two different approaches to key retrieval from the Java language: standard Java, or Java Native Interface (JNI). Standard Java access provides an interface that is fully implemented in the Java language. The JNI interface links the Java language to other language objects such as DLLs and shared Libraries. Java developers prefer standard Java solutions because they are simpler to install and use than JNI interfaces.

If the key retrieval interface involves complex packaging of the key information, the key management vendor often has no choice but to use the JNI interface. Simpler packaging allows for the use of the standard Java interface without the use of other language objects.

Java can present some challenges when implementing secure key retrieval using the SSL/TLS protocol. Java SSL/TLS is complex and involves the use of Java Key stores for certificate management. Be sure your key management vendor can provide you with sample Java source that demonstrates secure key retrieval.

Key Management in the Cloud

Public and private organizations want to take advantage of cloud-based solutions to reduce costs and improve business performance. Organizations should understand that the ultimate responsibility for the security of their data remains with them.

While the new key management services offered by some cloud providers will help some customers better protect their sensitive data in their clouds, it will not meet minimum standards and security requirements for many organizations, and it will lock customers and partners into a specific cloud platform. Selecting a key management system is the most important part of a cloud encryption strategy.

Summary

Deploying encryption and key management across the enterprise involves work on the part of application developers on each enterprise computing platform. This work can be easier or harder depending on the key management vendor’s dedication to appropriate implementations on each platform, ease of use of the key retrieval interface, and the availability of sample...
source code. Solutions which are difficult to deploy in business applications raise the costs of encryption and key management. In some cases implementation can cost more than the initial key management solution.

---

**Townsend Security**

Townsend Security creates data privacy solutions that help organizations meet evolving compliance requirements and mitigate the risk of data breaches and cyber-attacks. Over 3,000 companies worldwide trust Townsend Security’s NIST and FIPS 140-2 compliant solutions to meet the encryption and key management requirements in PCI DSS, HIPAA/HITECH, FISMA, GLBA/FFIEC, SOX, and other regulatory compliance requirements.

You can contact Townsend Security for an initial consultation at the following locations:

- **Web:** www.townsendsecurity.com
- **Phone:** (800) 357-1019 or (360) 359-4400
- **International:** +1 360 359 4400
- **Email:** info@townsendsecurity.com
- **Twitter:** @townsendsecure
Key Management Vendor Checklist
10 questions to ask your key management vendor

1) How would you describe the encryption key payload as retrieved from the key server? Is it simple or complex?

2) Is there a common key retrieval application interface on all platforms? What are the differences?

3) What platforms do you support for key retrieval? (Note any gaps in platform coverage for your company)

4) Do you provide working sample code for the platforms I need? (Windows, Linux, IBM i, IBM z)

5) Do you have a .NET Assembly for use with Windows applications?

6) Do you provide an IBM i service program for RPG and Cobol applications?

7) Do you have a Java key retrieval class and examples? Is it standard Java or JNI?

8) Are there additional end-point licensing fees?

9) Can you run your key management solution directly in the Microsoft and Amazon clouds?

10) Does the key management solution run as a native VMware virtual machine?

Once you have the answer to the above questions, it should be easier to choose the right key management vendor for your enterprise. If you have any questions not covered here, we would be happy to help you find the answers. Give us a call for a more complete needs assessment.