Key Management for Enterprise Data Encryption

Ulf T. Mattsson
Protegrity Corp.

Abstract

One of the essential components of encryption that is often overlooked is key management - the way cryptographic keys are generated and managed throughout their life. Since cryptography is based on keys which encrypt and decrypt data, your database protection solution is only as good as the protection of those keys. Security depends on several factors including where the keys are stored and who has access to them. When evaluating a data privacy solution, it is essential to include the ability to securely generate and manage keys. This can be achieved by centralizing all key management tasks on a single platform, and effectively automating administrative key management tasks, providing both operational efficiency and reduced management costs. Data privacy solutions should also include an automated and secure mechanism for key rotation, replication, and backup. The difficulty of key distribution, storage, and disposal has limited the wide-scale usability of many cryptographic products in the past. Automated key distribution is challenging because it is difficult to keep the keys secure while they are distributed, but this approach is finally becoming secure and more widely used. Standards for key-management have been developed by the government and by organizations such as ISO, ANSI, and the American Banking Organization (ABA). The key management process should be based on a policy. This paper will exemplify different elements of a suggested policy for a Key Management System used for managing the encryption keys that protect secret and confidential data in an organization.

Keywords: Key management, Database encryption, Security, Privacy, PCI, VISA CISP, GLBA, HIPAA.

1 Introduction

Best practices dictate that we must protect sensitive data at the point of capture, as it's transferred over the network (including internal networks) and when it is at rest. Protecting data only sometimes – such as sending sensitive information over wireless devices over the Internet or within your corporate network as clear text -- defeats the point of encrypting information in the database. It's far too easy for information to be intercepted in its travels so the sooner the encryption of data

occurs, the more secure the environment will be. A comprehensive encryption solution doesn't complicate authorized access to the protected information -- decryption of the data can occur at any point throughout the data flow wherever there is a need for access. Decryption can usually be done in an application-transparent way with minimum impact to the operational environment. Due to distributed business logic in application and database environments, organizations must be able to encrypt and decrypt data at different points in the network and at different system layers, including the database layer. Encryption performed by the database management system can protect data at rest, but more security oriented corporations will also require protection for data while it's moving between applications, databases and data stores. One option for accomplishing this protection is to selectively parse data after the secure communication is terminated and encrypt sensitive data elements at a very granular level (usernames, passwords, and so on.). Application-layer encryption and mature database-layer encryption solutions allow enterprises to selectively encrypt granular data into a format that can easily be passed between applications and databases without changing the data.

2 Issues with point solutions

A major problem with encryption as a security method is that the distribution, storage, and eventual disposal of keys introduce an expensive and onerous administrative burden. Historically, cryptographic keys were delivered by escorted couriers carrying keys or key books in secure boxes. An organization must follow strictly enforced procedures for protecting and monitoring the use of the key, and there must be a way to change keys. Even with all of these restrictions, there is always a chance that the key will be compromised or stolen. Even if there are standards developed for key-management it is still the most difficult part of an encryption solution. This is one of the greater challenges to overcome when you decide to create your own solution based on encryption toolkits from database vendors and security vendors. These toolkits provide the basic functionality for encrypting and decrypting information but typically do not provide a secure key-management system. Many companies have tried to develop their own encryption functionality, but few have succeeded in creating a system that performs not only by doing the obvious encryption, but doing so in a secure and reliable manner that does not prohibit you from keeping your systems operational. A mature data protection system should be based on a sophisticated key management system that is transparent, automated, secure and reliable for the environments where it operates.

3 A distributed approach with a central point of control

A mature data protection system should provide a central point of control for data protection systems at the application, database and file levels. The encryption solution has a combined hardware and software key management architecture which combine the benefits of each technology. This will address

the central security requirements while providing the flexibility to allow security professionals to deploy encryption at the appropriate place in their infrastructure. It provides advanced security and usability smooth and efficient implementation into today's complex data storage infrastructures. If your human resources department locks employee records in filing cabinets where one person is ultimately responsible for the keys, shouldn't similar precautions be taken to protect this same information in its electronic format? One easy solution is to store the keys in a restricted database table or file. But, all administrators with privileged access could also access these keys, decrypt any data within your system, and then cover their tracks. Your database security in such a situation is based not on industry best practice, but on trusting your employees. When securing the sensitive data within your organization trust is not a policy. The key custodian should be a role in the IT organization.

3.1 The key custodian

The key custodian is responsible for managing the multi-layer key management infrastructure, including the creation of keys, distribution of replacement keys and the deletion of keys that have been compromised. The custodian should be appointed by the Compliance Review Committee. Access to central key management functions should require a separate and optional strong authentication and management of encryption keys should be logged in an evidence-quality audit system. Keys stored in the Hardware Security Module are protected from physical attacks and cannot be compromised even by stealing the Hardware Security Module itself. Any attempt to tamper with or probe the Hardware Security Module will result in the immediate destruction of all private key data, making it virtually impossible for either external or internal hackers to access this vital information. Encryption of the application data should be performed by an Enforcement Agent that should be implemented as a Dedicated Encryption Service (Please see my articles in http://www.net-security.org/dl/insecure/INSECURE-Mag-8.pdf and http://www.revealnet.com/newsletter-v6/1105_B.htm) that is separated from the administration of the data that it protects. This service may run in different environments including in a separate process, a separate server or in a Hardware Security Module depending on the security class of the data and the operational requirements for performance and availability.

3.2 Key domains for added protection and easier management

A mature data encryption solution should support the concept of key domains which can isolate different systems for security reasons or operational needs. Each key domain may have different security exposures and can have a different policy for how keys should be managed including key generation, key rotation and protection of key material. It should support transparent re-encryption of the data when it flows between systems that are using different encryption keys or different algorithms. The Key Management System must support multiple levels of keys to ensure that the encryption keys that protect secret and confidential

data cannot be compromised. This enables the use of different encryption keys for different columns, tables and files. When setting policy, it is important to configure the use of different encryption keys and initialization vectors across different columns, tables and files to maintain compartmentalization and a diverse front against attack. The Keys should be stored in an Enforcement Agent that supports dual control (requiring more than a single administrator/operator) for key recovery. It may be implemented in hardware or software, but it must support both the encryption and integrity of the key backup format.

4 Annual review of algorithms and key lengths

The Key Management System must support key length or strength of 128-bits or greater for symmetric keys. Such keys are deemed "strong encryption" and are not susceptible to a brute force attack using current technology. Public or asymmetric keys must be of equivalent strength. That is, a 128-bit symmetric key and 3072-bit public key are considered to be equivalent in terms of strength, while a 15,360-bit public key is equivalent to a 256-bit symmetric key. The data encryption should be performed with strong standard algorithms including 3DES, AES 128 or AES 256. Data requiring protection for longer periods of time should use the longer key lengths. Note that adequate CPU power today may not be enough tomorrow as you incorporate more secure communications. It is wise to establish a key-length policy early and review it annually.

5 Secure generation and distribution of keys

The Key Management System must generate a unique key for each file, tape, or other data element that needs to be encrypted. Private keys must be generated within the secure confines of the Key Management System and never be transferred outside the Key Management System unless encrypted with a Key Encryption Key. All keys should be centrally generated in software or hardware based on the security class for the type of data they protect. The key management system must be able to electronically transfer private keys to other trusted key repositories throughout the enterprise. This may also be implemented via Smart Cards. The security policy should define where different keys should be stored and cached. The master keys are used to encrypt all operational keys that should be stored in cipher text in separated databases. Security metadata and operational encryption keys should be kept in cipher text (even when stored in memory) until needed for use by crypto-services routines. All communication both external and internal is encrypted. All Data Protection System services should be using X.509 certificates and SSL for secure distribution of encryption keys. Unique keys should be generated for each Enforcement Agent, and should be used when sending information between system components. The data encryption method should be based on different encryption keys for different columns, tables, files and directories. An optimal design for Hardware Security Module support can be based on an optimal combination of hardware and software keys. Supported Hardware Security Module should be tamper evident

and compliant with FIPS PUB 140-2 Level 3 Security Requirements for Cryptographic Modules, and keys are randomly generated in compliance with ANS X9.24 Section 7.4.

5.1 Key validation, access control and logging

Key validation is performed by integrity checking the security metadata that is kept in ciphered text (even in memory). Key access control is performed by role-based authorization of users, allowing for specific authorized actions by user (select/insert/update/delete). Users can be authenticated by any accepted means of the native database. Any encrypt/decrypt operation requested by the user is verified against the policy by the Enforcement Agent after authorization and authentication checks have been completed by the database. Under the control of the authenticated Security Administrator, the system should generate a Master Key used to encrypt all operational keys. Security data remains ciphered until needed for use by crypto-services routines. The master keys and data encryption keys should be secured, and their integrity checked. All communication, external and internal, should be encrypted. The system may use public key cryptography to exchange the symmetric encryption keys. The Key Management System must support tracking of; when keys are created and deleted; who created and deleted them; who used what keys; and what was done with the key.

6 Key protection and aging

Encryption keys should be protected and encrypted when stored in memory or databases, and during transport between systems and system processes. The use of a combination of software cryptography and specialized cryptographic chipsets, called a Hardware Security Module, can provide a selective added level of protection, and help to balance security, cost, and performance needs. Certain fields in a database require a stronger level of encryption, and a higher level of protection for associated encryption keys. Encryption keys and security metadata should continuously be encrypted and integrity validated – even when communicated between processes, stored or cached in memory. Security data should remain ciphered until needed for use by crypto-services routines. Key Rotation, or more accurately Key Aging, is best security practices and required in some governmental regulations and industry initiatives. More sensitive data and data more exposed systems should be re-encrypted with fresh encryption keys more frequently than the rest of the data. A well designed automated key rotation solution can provide zero down-time by attaching key labels to each record or data field in the operation databases and file systems. The Automated key rotation process can run in background and utilize spare cycles on each available processor on your data servers. The background processing can be assigned a priority level that will complete the key rotation according to the policy that is defined.

6.1 Secure key storage

To maintain a high level of security the end-point server platform should provide the choice to only temporarily cache encrypted lower level data encryption keys. Key encryption keys should always be stored encrypted on separated platforms. A central server with a hardened standard computing platform to store the keys can provide a cost effective solution. Keys should be kept in an encrypted format in memory (cached) until they are to be used. Data encryption keys should be stored in encrypted format in a separate data server along with other policy information, optionally on the Security Administration System repository or on the local database where the Enforcement Agent is installed, depending on the operational requirements and security level of the data that is protected. All keys except the Master Key should be stored (encrypted) under the Key Encryption Keys. The Master Key should also be protected while in transient storage or be kept inside the Hardware Security Module storage, depending on the operational requirements and security level of the data that is protected by the keys.

6.2 Protection of memory cached keys

Memory attacks may be theoretical, but cryptographic keys, unlike most other data in a computer memory, are random. Looking through memory structures for random data is very likely to reveal key material. Well made libraries for use as Native Encryption Services go to great efforts to protect keys even in memory. Key-encryption keys are used to encrypt the key while it is in memory and then the encrypted key is split into several parts and spread throughout the memory space. Decoy structures may be created to mimic valid key material. Memory holding the key is quickly zeroed as soon as the cryptographic operation is finished. These techniques reduce the risk of memory attacks. Separate encryption keys should be used for different data. These encryption keys can be automatically rotated based on the sensitivity of the protected data. A Dedicated Encryption Systems can provide separation between processes or servers dedicated to encryption operations but they are also vulnerable to memory attacks. However, a well made Dedicated Encryption System runs only the minimal number of services. Since web servers, application servers, and databases have no place on a dedicated cryptographic engine, these common attack points are not a threat. This severely constrained attack surface makes it much more difficult to gain the access needed to launch a memory attack. The security classification of the protected data will help in deciding a topology that will give the right balance between security, performance and scalability for each type of environment within an organization.

7 Key backup and recovery

A weak link in the security of many networks is the backup process. Often, private keys and certificates are archived unprotected along with configuration data from the backend servers. The backup key file may be stored in clear text or

protected only by an administrative password. This password is often chosen poorly and/or shared between operators. To take advantage of this weak protection mechanism, hackers can simply launch a dictionary attack (a series of educated guesses based on dictionary words) to obtain private keys. To maintain a high level of security and separation the application data backup files should be separated from the backup of encrypted lower level data encryption keys. After keys are created, they must be archived to a secure storage environment where they can be kept for long periods of time. Master keys should be backed up separately. During installation, the master key should be generated and stored on removable media for recovery purposes. Maintaining this media in escrow and/or at your disaster recovery site is best practice. Backup of keys on the Security Administration System should be performed on a regular basis, usually before and after major policy changes are realized. Backup of the encrypted data encryption keys should be automated and performed at the same time as business data backup, using standard database backup and restore procedures. Even if policies or keys have changed, or if the Security Administration System is unavailable, any Enforcement Agent and its protected database may be restored successfully as long as access to the Master Key is provided via proper user authentication. The Key Management System must be able to survive multiple hardware and site failures and still be able to retrieve the archived keys to unlock encrypted data. The Key Management System must support creation and management of "split keys," so that the ability to decrypt data requires cooperation of multiple persons, each knowing only their part of the key, to reconstruct the whole key.

Conclusion

We have reviewed crucial guidelines and best practices for a Key Management System for data encryption based on the approach of a central point of control for key management and distributed encryption and policy enforcement across applications, databases and file systems. The solution provides great flexibility by combining the benefits from hardware and software based encryption and key management. This approach addresses the requirements for central security control while providing the flexibility to allow security professionals to deploy encryption at the appropriate place in their infrastructure. It provides the needed balance between advanced security, availability, and performance for the combined solution. The concept of separate key domains across a data flow can isolate different systems from a risk perspective and it can also accommodate for differences in the operational requirements. Best practices dictate that we must protect sensitive data at the point of capture, as it's transferred also over internal networks and when it is at rest. A mature solution for encryption and key management can provide this higher level of protection of information.