

Implementing Data Privacy in the Enterprise: Best Practices in Key Management

By Derek Tumalak

Drivers for Data Privacy, Encryption, and Key Management

In spite of a range of security technologies being deployed, devastating thefts of sensitive data continue to occur. These ongoing breaches illustrate that organizations need to implement data privacy inside the enterprise. While traditional technologies like firewalls and intrusion detection systems are a critical part of protecting an enterprise's network perimeter, they are only part of a complete security picture. Recent research illustrates a few reasons for these security shortfalls:

- ▲ According to Gartner, 75% of external-based attacks are tunneling through applications—and so go undetected by a range of perimeter security mechanisms.
- ▲ The ongoing battle of patching known exploits is being lost: According to a study by Symantec, in 2003, 70 percent of all security vulnerabilities were simple for attackers to manage, and this number grew 10% over the previous year.
- ▲ Most estimates cite that now over 50% of security breaches are perpetrated by internal staff.
- ▲ Even with a fortified network perimeter, storage systems can be breached via insecure storage management interfaces and physical storage systems and data in the databases and applications themselves can be stolen.

Failure to address the security gaps evidenced above can have a disastrous effect on an organization. For years now, the price organizations have paid when breaches become public has been catastrophic. The enactment of policies and legislative mandates are not only dictating a more data-centric approach to security, but are requiring the disclosure of any breaches—and in many cases dramatically exacerbating the costs with civil litigation and fines. These mandates are coming in a range of forms:

- ▲ Regional legislation. Europe's Data Privacy Act, Canada's Personal Information Protection and Electronic Document Act (PIPEDA), California's Database Security Breach Notification Act, SB 1386, and many others all dictate encryption in some fashion, and that any victims of breaches are notified.
- ▲ Industry-specific legislation. In health care, the Health Insurance Portability & Accountability Act (HIPAA); and the Gramm-Leach-Bliley Act (GLBA) in financial services have provided comprehensive guidelines for safeguarding patient and consumer data respectively.
- ▲ Commerce policies. Credit card issuers like Visa, MasterCard, and American Express all have delivered comprehensive guidelines that

provide an edict for both best security practices, including data encryption for example, as well as mandating consumer notification of breaches.

To effectively address these changing dynamics and ensure data privacy, information security managers need to secure critical data as it is being stored, transmitted, and used within their organization's networks. A data privacy implementation consists of several building blocks, including cryptography, secure key management, cryptographic operations, authentication and authorization, logging and auditing, backup and recovery, and more. This article will focus on one of the most critical of these building blocks: key management.

The Importance of Effective Key Management

Encrypting data as it is being stored and transmitted inside the enterprise is one of the central elements to achieving data privacy inside an enterprise. However, encrypting data can be complex and time-consuming—and if it isn't done right, it may not adequately address fundamental security gaps. While encryption can help prevent data theft, ultimately your data is only as safe as the keys that protect it. Whoever has access to the keys has access to your data. And once an attacker has the key, it's relatively easy to copy, modify, delete, hijack, or destroy sensitive information, including credit card numbers, patient information, insurance data, and confidential documents.

Server Vulnerabilities

Today, in many cases, if organizations are deploying encryption on data to be stored on back-end servers and databases, the cryptographic keys are often stored on the same systems.

This strategy can introduce several security vulnerabilities. In large part, this is because Web application, and database servers are not dedicated security appliances—often they are relatively easy to access, their security capabilities can be disabled or misconfigured, and they aren't kept up to date with the latest security patches. They were never intended to function as security platforms, which makes them easy prey for cyberattackers inside or outside the organization. When cryptographic keys are stored on unsecured platforms, smart attackers can gain access to them very quickly because they are often stored in an easily readable plain text format. And as more keys are stored on servers, it becomes even easier to locate and manipulate them.

Even organizations that make efforts to protect cryptographic keys with passwords find that these passwords are poorly chosen, poorly protected,

and usually must be shared between multiple administrators—all of which increases the vulnerability of secret keys.

This type of server-based approach to key management can also become prohibitively costly, both from a budgetary and resources perspective. Purchasing multiple cryptographic cards for each server can quickly add up, and administering these cryptographic resources in a disparate fashion is very time consuming.

Secure Key Architectures

To achieve a highly secure environment, information security managers will want to generate and manage keys in a centralized manner in which strict access privileges are enforced. For example, keys stored across multiple application server and database hard drives are significantly more difficult to manage and protect than keys stored on a centralized platform.

Hardware

A specialized hardware device in which all cryptographic operations are performed securely and in which keys are never visible in the clear is highly recommended. This provides a significantly higher level of security over a pure software solution in which keys are managed and used in the clear. Some highly specialized hardware can provide a level of tamper resistance, so that, if an attempt is made to compromise the keys, the hardware will clear all information including the keys. This type of hardware solution is recommended for enterprises that require an extremely high level of security.

Secure Key Processes

There are several important issues to consider in terms of the process of key management inside an organization. Following are a few critical issues:

Import

Importing a key into a secure key management system is not recommended since the system has no way of verifying where the key has existed prior to the import or even if the key has been compromised. If key import is a requirement, the history of the key should be well documented, and all copies of the key should be managed carefully.

Export

Exporting a key from a secure management system is not recommended since the system will have no means of verifying how the key is used once it leaves the secure environment. If key export is a requirement, exported copies of the key should be managed carefully.

Rotation

It is good practice to protect data with newly generated keys periodically. Re-encrypting data with a new key at least once a year is recommended. An important consideration when rotating keys is managing backups and archives. An enterprise must be able to ensure that sensitive data cannot be compromised through the use of old keys and archived data, while also being able to guarantee access to this data if necessary.

Secure Key Administration

Controlling access to cryptographic keys is another essential component to secure key management. As mentioned above, centralizing key

management is one of the best ways to secure access to keys. If key management is done on a centralized platform, organizations should ensure the platform can only be accessed at the administrator level, via a secure connection. The platform should also allow for capabilities for splitting administrative access so that individual administrators are granted access only to areas for which they are responsible. For example, one administrator might only be given access to network configuration, while another might only be given access to certificate management. This level of granular access control enables customers to control and closely monitor administrator operations. All actions performed by all users and administrators should also be securely logged, stored, and available for reporting purposes.

In addition, some platforms support smart cards or other physical administrative safeguards. Smart cards enable multiple administrators to share parts of a “group key”, which would be required to perform sensitive administrative functions—such as backup and restore operations. This additional level of security makes it impossible to corrupt the system if a smart card is lost or stolen because multiple smart cards must be used together to gain platform access.

Key Management for Database Encryption

When encrypting data for databases, there are several issues that information security managers must consider from the standpoint of key management.

Encryption of Multiple Columns

If multiple columns of a database table are encrypted, it is strongly recommended to use different encryption keys for each column. That way, even if an attacker manages to compromise a single key, the rest of the encrypted columns will remain secure. The only reason to use a single key to encrypt multiple columns is if the columns all contain values from the same set of data and the encrypted values have to be compared with each other to determine equality (such as when performing a join). The database schema and application logic should be designed so as to minimize situations where this is necessary.

Indexes

Indexes are created to facilitate the search of a particular record or a set of records from a database table. Indexes are created on a specific column or a set of columns. When the database table is selected, and WHERE conditions are provided, the database will typically use the indexes to locate the records, avoiding the need to do a full table scan. In many cases searching on an encrypted column will require the database to perform a full table scan regardless of whether an index exists. For this reason, encrypting a column that is part of an index is not recommended.

Primary Key

Encrypted columns can be a primary key or part of a primary key, since the encryption of a piece of data is stable (i.e., it always produces the same result), and no two distinct pieces of data will produce the same ciphertext, provided that the key and initialization vector used are consistent. However, when encrypting entire columns of an existing database, depending on the data migration method, database administrators might have to drop existing primary keys, as well as any other associated reference keys, and re-create them after the data is encrypted. For this reason,

encrypting a column that is part of a primary key constraint is not recommended. Since primary keys are automatically indexed there are also performance considerations, as described above.

Foreign Key

A foreign key constraint can be created on an encrypted column. However, special care must be given during migration. In order to convert an existing table to one that holds encrypted data, all the tables with which it has constraints must first be identified. All referenced tables have to be converted accordingly. In certain cases, the referential constraints have to be temporarily disabled or dropped to allow proper migration of existing data. They can be re-enabled or recreated once the data for all the associated tables is encrypted. Due to this complexity, encrypting a column that is part of a foreign key constraint is not recommended. Unlike indexes and primary keys, though, encrypting foreign keys generally does not present a performance impact.

Initialization Vectors

When using CBC mode of a block encryption algorithm, a randomly generated initialization vector is used and must be stored for future use when the data is decrypted. Since the IV does not need to be kept secret it can be stored in the database. If the application requires having an IV per column, which can be necessary to allow for searching within that column, the value can be stored in a separate table. For a more secure deployment, but with limited searching capabilities, an additional column can be added to the table and an IV can be generated per row. In the case where multiple columns are encrypted, but the table has space limitations, the same IV can be reused for each encrypted value in the row, even if the encryption keys for each column are different, provided the encryption algorithm and key size are the same.

Searching

Searching for an exact match of an encrypted value within a column is possible, provided that the same initialization vector is used for the entire column. On the other hand, searching for partial matches on encrypted data within a database can be challenging and can result in full table scans. One approach to performing partial searches, without prohibitive performance constraints and without revealing too much sensitive information, is to apply an HMAC to part of the sensitive data and store it in another column in the same row. For example, a table that stores encrypted customer email addresses could also store the HMAC of the first four characters of each email address. If a customer service representative wanted to search for all customers whose email addresses began with "john" the system would apply an HMAC on "john" and search through the HMAC column for matches, without having to decrypt every single full email address. This approach can be used to find exact matches on the beginning or end of a field (e.g., "john", "yahoo.com").

One drawback to this approach is that a new column needs to be added for each unique type of search criteria. So if the database needs to allow for searching based on the first four characters as well as the last five characters, two new columns would need to be added to the table. However, in order to save space, the HMAC hash values can be truncated to ten bytes without compromising security. This approach can prove to be a reasonable compromise especially when combined with non-sensitive search criteria such as zip code, city, etc. and can significantly improve search performance.

Encoding

If data is to be managed in binary format, varbinary can be used as the data type to store encrypted information. On the other hand, if a binary format is not desirable, the encrypted data can be encoded and stored in a varchar field. There are size and performance penalties when using an encoded format, but this may be necessary in environments that do not interface well with binary formats.


Available Space

In environments where it is unnecessary to encrypt all data within a database, a solution with granular capabilities is ideal. Even if only a small subset of sensitive information needs to be encrypted, additional space will still be required. Make sure that enough space exists to accommodate new fields, metadata, as well as the temporary space that will be required to perform the data migration.

Pre-Migration Backups

Even if sensitive information in production databases is securely protected, it is important to be aware that sensitive data may still exist in the clear in such places as tape backups and database backups. An enterprise must identify all of these locations and replace them with new backups in which the sensitive information is protected.

Key Management for Successful Security

With the continued increase in security breaches, assessing a network's security needs is an ongoing task for any information security manager. This article has provided a quick introduction into some of the ideas, approaches and issues surrounding encryption and key management, but there is much more to a security assessment. Your investment in ensuring that you have the right strategy for encryption and key management may be the ounce of prevention that your company needs to protect its most important information and ensure data privacy. 

Derek Tumalak is director of product management at Ingrian Networks (www.ingrian.com) where he also served as an engineering manager focused on the development of security solutions for more than six years, developing a secure payment solution in France and bringing E-Stamp's electronic postage solution to market. Tumalak earned his degree in Computer Engineering at the University of Waterloo in Ontario, Canada.