

# ***Data Privacy in the Enterprise (Best Practice)***

## ***1. Encryption***

This section provides a list of recommendations for choosing among the various cryptographic operations available in implementing a data privacy solution.

### ***1.1 Data Encryption Standard (DES)***

DES encrypts and decrypts data in 64-bit blocks, using a 64-bit key (although the effective key strength is only 56 bits). It takes a 64-bit block of plaintext as input and outputs a 64-bit block of ciphertext. Since it always operates on blocks of equal size and it uses both permutations and substitutions in the algorithm, DES is both a block cipher and a product cipher.

DES has 16 rounds, meaning the main algorithm is repeated 16 times to produce the ciphertext. It has been found that the number of rounds is exponentially proportional to the amount of time required to find a key using a brute-force attack. So as the number of rounds increases, the security of the algorithm increases exponentially.

Although DES has enjoyed widespread popularity for many years, it is no longer considered secure enough by today's standards. DES should generally not be used in production environments.

### ***1.2 Triple Data Encryption Standard (TDES)***

Triple DES is simply another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits. In Private Encryptor, you simply type in the entire 192-bit (24 character) key rather than entering each of the three keys individually. The Triple DES DLL then breaks the user provided key into three subkeys, padding the keys if necessary so they are each 64 bits long. The procedure for encryption is exactly the same as regular DES, but it is repeated three times. Hence the name Triple DES. The data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key.

## ***Data Privacy in the Enterprise (Best Practice)***

Consequently, Triple DES runs three times slower than standard DES, but is much more secure if used properly. The procedure for decrypting something is the same as the procedure for encryption, except it is executed in reverse.

3DES is a more secure alternative to DES and is a widely used symmetric algorithm. If possible, AES should be used rather than 3DES due to the performance advantages offered by AES.

### ***1.3 Advanced Encryption Standard (AES)***

The Advanced Encryption Standard (AES) is an encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each AES cipher has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide

AES is a newer and preferred choice today and is gaining broad acceptance in the industry due to its performance advantages over 3DES.

### ***1.4 Rivest Cipher 4 (RC4)***

The RC4 works on an array of 256 bytes to generate a pseudo random number sequence which is used as keystream to encrypt data.

RC4 should not be used for data protection but rather for encryption during short-lived sessions. A single key should never be used to encrypt more than one piece of data.

### ***1.5 Message Digest 5 (MD5)***

Message Digest 5 is a standard algorithm that takes as input a message of arbitrary length and produces as output a 128-bit fingerprint or message digest of the input. Any modifications made to the message in transit can then be detected by recalculating the digest.

### ***1.6 Secure Hash Algorithm (SHA1)***

## ***Data Privacy in the Enterprise (Best Practice)***

The US Secure Hash Algorithm takes a message of less than 264 bits in length and produces a 160-bit message digest designed so that it is computationally very expensive to find a text string that matches a given hash.

For a number of years, MD5 has been a favorite hashing algorithm, but recent efforts have revealed certain attacks that weaken MD5's protection. In general, SHA-1 should be used over MD5 for hashing data.

### ***2. Key Management***

Key management is a fundamental consideration when deploying a data privacy solution. If the keys used to protect sensitive data within an enterprise are not properly secured, attackers may be able to gain access to this data with relative ease. This section discusses some of the considerations when managing keys in the context of enterprise data privacy.

#### ***2.1 Centralizing Storage and Administration***

In a highly secure environment it is important 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.

#### ***2.2 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.

#### ***2.3 Import***

## ***Data Privacy in the Enterprise (Best Practice)***

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.

### ***2.4 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.

### ***2.5 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.

## ***3. Authentication, Authorization, and Auditing***

Enterprises need a secure way to identify people and entities that require access to sensitive data. In implementing a solution, administrators need to decide what data will be accessible and who will have access to it. Some methods of access control are passwords, client certificates, biometrics, and tokens.

### ***3.1 Authentication***

Authentication ensures that an entity is really what or who it says it is. Methods of authentication can be classified as what you know, what you have, or what you are. The traditional “what you know” form of authentication, a username and password, is the least secure. Password can be given away, guessed, or

## ***Data Privacy in the Enterprise (Best Practice)***

stolen. “What you have” is called a token such as a client certificate or a smart card. “What you are” can be proven by recording and comparing a voiceprint, fingerprint, retinal scan, or even DNA. In general, the more factors used for authentication, the stronger the authentication is. For example, a system that uses a username and password along with a client certificate provides a higher level of security than one that only uses a username and password.

### **3.2 Authorization**

Authorization ensures that only the entities who should have access to resources obtain that access. In order for authorization to be granted to an entity, it must first be authenticated. Two of the authorization methods available are rolebased security and subject/object access control. Role-based security means that authorization is defined based on an entity’s responsibilities. For example, an enterprise may choose to create an “application” role that only has authorization to encrypt credit card numbers and a “processing” role that only has authorization to decrypt the same information. In a subject/object access control system, every resource (“object”) has an explicit list of entities (“subjects”) that are allowed access. The least privilege security principle states that entities should only be granted the absolutely minimal set of privileges necessary to perform their tasks. Additional unnecessary authorization privileges only increase the vulnerability of the system to attack. It is important to be wary of “access creep”, where employees are granted more and more authorization over time by special request. It is also critical to remember to change an employee’s role when he or she changes jobs.

### **3.3 Auditing**

Auditing is an extremely important part of a data privacy solution. It allows the enterprise determine who did what at any given point in time, including when authentication and authorization were allowed or denied to an entity. A data privacy solution should offer robust logging capabilities and support log signing, in order to prevent an attacker from tampering with logs. Logs should be

## ***Data Privacy in the Enterprise (Best Practice)***

analyzed regularly to look for strange behavior that could potentially represent attacks on the enterprise.

### ***3.4 Credentials***

One of the challenges related to authentication is the ability to properly secure the credentials used to authenticate. Usernames and passwords, if stored on a local machine, should not be kept in the clear. At the very least, some form of encoding or obfuscation should be performed so that this information is not easily humanly readable. Certificates should be stored in a key store that is password protected.

## ***4. Backup, Restore, and Disaster Recovery***

Backup and restore capabilities are critical to ensure that an environment can be recreated in the event of a disaster. It is also important to be able to replicate an existing environment in order to scale according to the needs of the enterprise. A good solution will allow for a secure mechanism to create backups and perform restores of all keys as well as relevant configuration information. In some environments, specialized hardware used to protect keys has its own mechanism for backup and restore.

## ***5. Load Balancing and High Availability***

### ***5.1 Load Balancing***

A load-balanced system is a good way to provide a highly redundant environment and scale performance to meet the needs of the enterprise. A data privacy solution should be able to load balance intelligently across different physical locations as well as perform monitoring services to determine if systems are up or down.

### ***5.2 High Availability***

In some enterprise environments, it may be desirable to have an active/passive model in which one machine actively services requests while

## ***Data Privacy in the Enterprise (Best Practice)***

another stands by. In the event that the active system fails, the standby system assumes the IP address of the previously active system and starts to service requests.

### ***5.3 Replication***

Deployments with multiple systems can often be difficult to maintain. Platforms that offer automated configuration and key replication services are more desirable. This simplifies the overall management experience and reduces the chances of errors.

## ***6. Network and Transport***

### ***6.1 Transport***

It is recommended to use SSL at all points in which sensitive data is in transit, both over the Internet and within the enterprise (such as between the application server and the database). A good data privacy solution will allow for secure transmission of sensitive data between all entities across an enterprise.

### ***6.2 Firewall***

In general, all administrative ports should be blocked from external networks. Access control lists (ACLs) should be set up to restrict access to certain devices on the network. For example, if a network-based security appliance resides on the network, only those devices that require access for administrative or cryptographic operations should be granted access.

## ***7. Database Considerations***

### ***7.1 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

## ***Data Privacy in the Enterprise (Best Practice)***

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.

### ***7.2 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.

### ***7.3 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.

### ***7.4 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

## ***Data Privacy in the Enterprise (Best Practice)***

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.

### ***7.5 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.

### ***7.6 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 Hash Message Authentication Code (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

## ***Data Privacy in the Enterprise (Best Practice)***

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 in order to save space. 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.

### ***7.7 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.

### ***7.8 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.

### ***7.9 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

## ***Data Privacy in the Enterprise (Best Practice)***

identify all of these locations and replace them with new backups in which the sensitive information is protected.

[www.YouSigma.com](http://www.YouSigma.com)