# Transparent Database Encryption in SQL Server: A Planning Guide

Ron Johnson

## Introduction

IT security is, generally, defined as a defensive approach to protect a company and its assets from unauthorized access by an intruder. IT security efforts include network security appliances, HoneyPots, robust authentication, limiting authorization to least necessary privileges, as well as other perimeter security defenses. However, these approaches do not provide definitive protection of the company's most valuable asset, its data, because a single intrusion could result in sensitive data being compromised. Additionally, in today's workplace culture the disgruntled employee may be as much of a threat as any external threat.

Data encryption is a direct response to internal and external security threats that may also meet compliance regulations. Encryption provides strong security for data "at-rest"; in our case, the data stored in the database, but to be effective should be implemented as a part of a broader security plan. There are many issues involved with the implementation of encryption, details that require decisions and actions to ensure the success of the implementation and the security of the data. This document will discuss the issues associated with database encryption implemented using SQL Server's native Transparent Database Encryption (TDE) mechanism.

## Encryption Overview

Encryption has been integral to human history beginning with the Babylonian use of Intaglio other historical examples include the Caesar Cipher, Scytale Transposition Cipher, Enigma, and even Jim Sanborn's Kryptos sculpture. Throughout history our society has enjoyed the ability to protect information using cryptographic methods including steganography, microdots, invisible ink, digital watermarks, and encryption which may be defined as the conversion of data so as to keep its meaning private. As the amount of sensitive data collected by commercial entities continues to grow the regulatory requirements for protecting the sensitive data will become more robust; meeting the regulatory requirements will necessarily require the continued use of data encryption methods.

Encryption requires the application of an algorithm to transform the target data into a form that is unusable to anyone that does not have access to the encryption process used. In practical terms encryption applies a cryptographic algorithm with a "key" to the target data producing the encrypted form of the data which cannot be accessed without the key used to encrypt the data. The two primary forms of key encryption are symmetric and asymmetric which are distinguished by the number of keys used in the encryption / decryption process. Symmetric encryption uses a single key while asymmetric encryption uses a pair of keys generally referred to as public and private keys.

While asymmetric encryption appears ideal for implementation because only the public key need ever be shared there are disadvantages with regard to performance. A sampling of asymmetric algorithms includes RSA, DSA, ELGamal, ECDSA, and XTR. Figure 1 demonstrates the asymmetric encryption process.

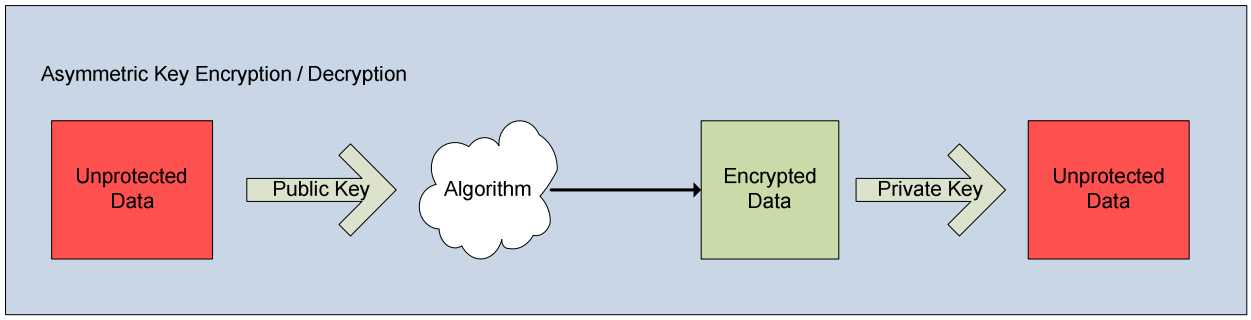


Figure 1 Asymmetric Key Encryption / Decryption Process

Symmetric algorithms require a single key for both encryption and decryption which allows for high-performance; however, with this approach the strength of the encryption is dependent on the security of the key. Common symmetric algorithms include AES/Rijndael, Blowfish, DES, Triple DES, Serpent, and IDEA to name only a few. Figure 2 demonstrates the symmetric encryption process.

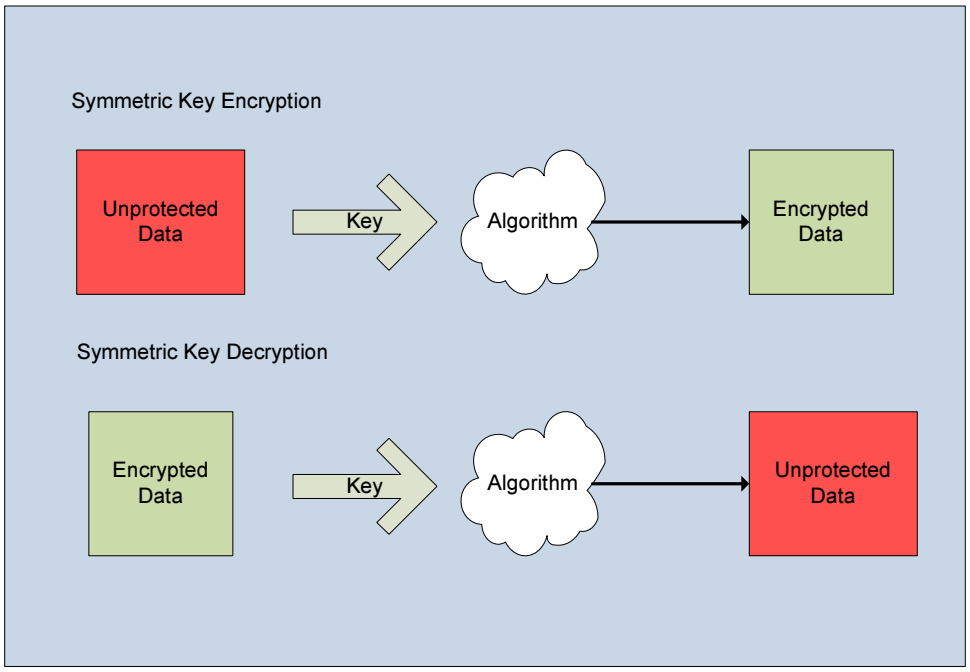


Figure 2 Symmetric Key Encryption Process

Both symmetric and asymmetric encryption approaches are vulnerable to brute force attacks and cryptanalysis. Brute force is an attack during which every possible permutation of the key value is attempted. Cryptanalysis, on the other hand, applies computational techniques to circumvent the encryption. In general, the use of sufficiently long keys will mitigate these attacks.

In summary, a symmetric key algorithm is fast but less secure than an asymmetric algorithm. Another approach is a hybrid wherein a symmetric key is used to encrypt the data while an asymmetric key is used to encrypt the symmetric key. It may be important to know in order to maintain perspective that there is only one encryption algorithm that is impossible to crack, One-Time Pad (OTP), any other algorithm may be broken given sufficient time and / or computer resources.

Security concerns, in general, and encryption, specifically, are new concepts for most IT professionals; therefore, a Glossary of Security / Encryption Terms is included as an appendix for reference.

## Overview of Transparent Database Encryption

The primary benefit of Transparent Database Encryption (TDE) is the ability to encrypt data without affecting any application that uses the data while providing security for the entire database. TDE is implemented at the database-level, unlike cell-level encryption TDE does not require modification to applications or database column data types; furthermore, database-level encryption allows for higher performance than cell-level encryption. However, TDE may allow more data leakage because encrypted data is decrypted when read into the buffer pool; therefore, the data is not protected if the operating system writes data from memory to disk during paging operations, or during hibernation, or memory dumps, nor is the data protected while in memory.

Database encryption is achieved by leveraging the Data Protection API (DPAPI) in Windows® which protects the Service Master Key (SMK) which protects the Database Master Key (DMK) which is used to protect the certificate or asymmetric keys which are used to protect the Database Encryption Key (DEK). These dependencies create a security chain from the operating system to the data eliminating user interaction thus strengthening security. The relationships and dependencies between keys is represented in Figure 3 below:

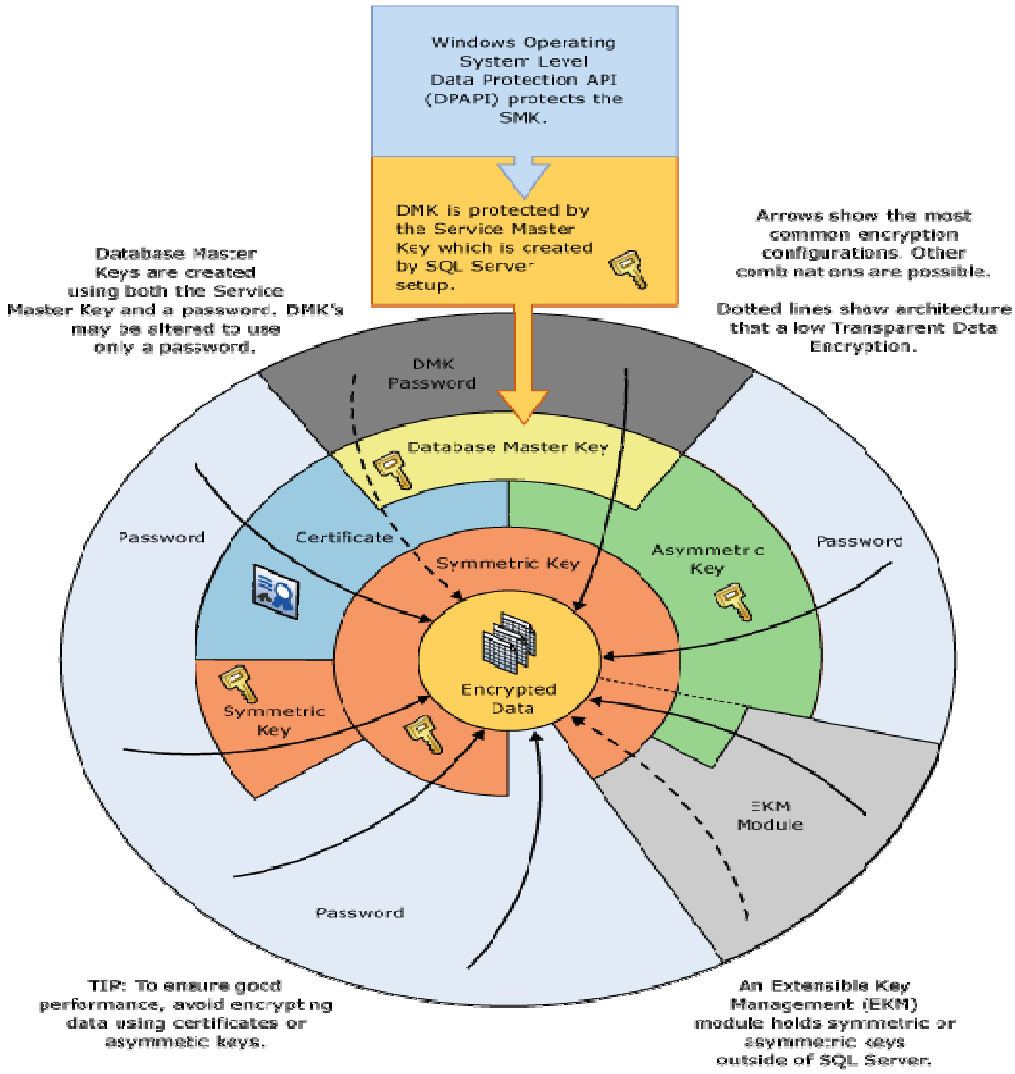


**Figure 3 SQL Server encryption key hierarchy with TDE and EKM (Source: BOL - http://msdn.microsoft.com/en-us/library/cc278098.aspx)**

The hierarchy of keys in TDE is protected from the DPAPI to the DEK allowing the server to manage encryption and decryption automatically. The DMK and the certificate are stored in the *MASTER* database while the DEK is stored in the user database. This hierarchy and the key management chain provide TDE the capability to transparently encrypt and decrypt the database.

The process for encrypting a database is conceptually simple:

1. Create a Master Key
2. Obtain an Authentication Certificate
3. Create DEK
4. Enable TDE on the database

However, significant complexity will be introduced if the database encryption strategy is undertaken without proper planning that addresses important implementation issues. Those issues are discussed in the following section.

## Encryption Issues

The level of security necessary to protect the database should be documented during the planning phase. Individually and in combination the following encryption mechanisms are available to secure the database:

- Encrypting File System (EFS)
- Cell-level
- BitLocker
- Transparent Database Encryption (TDE)

Discussion of the benefits and performance implications of each mechanism and their combinations is beyond the scope of this paper.

Data encryption must address two equally important issues: encryption technology and cryptographic key (*key*) management. Encryption technology provides for variable granularity of data protection, performance, and integration with existing applications, as well as ease of implementation and management. However, the success of the selected encryption strategy may depend most on key management policies and processes. Key management issues include: key access, key storage, and cryptographic algorithm. Key management is one of many important issues that must be considered when planning the encryption project.

The important issues to consider during the planning phase of the encryption project are listed below:

- **Encryption Algorithm**
  - DES, Triple DES, TRIPLE_DES_3KEY, RC2, RC4, 128-bit RC4, DESX, 128-bit AES, 192-bit AES, and 256-bit AES
- **Key Management**
  - Key Storage
    - Hardware Security Module (HSM)
    - SQL Server
  - Key Scheduling
  - Key Availability / Mobility / Security
- **Performance Impact**
  - Encryption / Decryption – Microsoft claims 3-5%; however, independent tests indicate 6-12%.
  - TempDB Encryption – Encryption of any one DB will encrypt TempDB.
  - Transaction Log is encrypted.

- **Log Shipping Implementation Changes**
  - Encrypted database log shipping requires the recipient database to possess the key in order to apply the logs.
- **Backup and Recovery Plan Changes**
  - Encrypted databases cannot be recovered to a different instance without the key.

- **Disaster Recovery Plan Changes**
  - Encrypted databases cannot be recovered to a different instance without the key.
- **Increased Disk Space Requirements**
  - No SQL Server native backup compression. Third party tools may be available; however, in general, encrypted data cannot be significantly compressed.
- **TDE operates during I/O; therefore, any data written to disk outside of the buffer pool is not protected**
  - No Support for FILESTREAM data-type

The diagram in Figure 4 represents a nominal encryption project planning process with each major area of consideration represented. The end result of the planning process is to produce a document detailing the decisions made that address the issues related to encrypting the database.

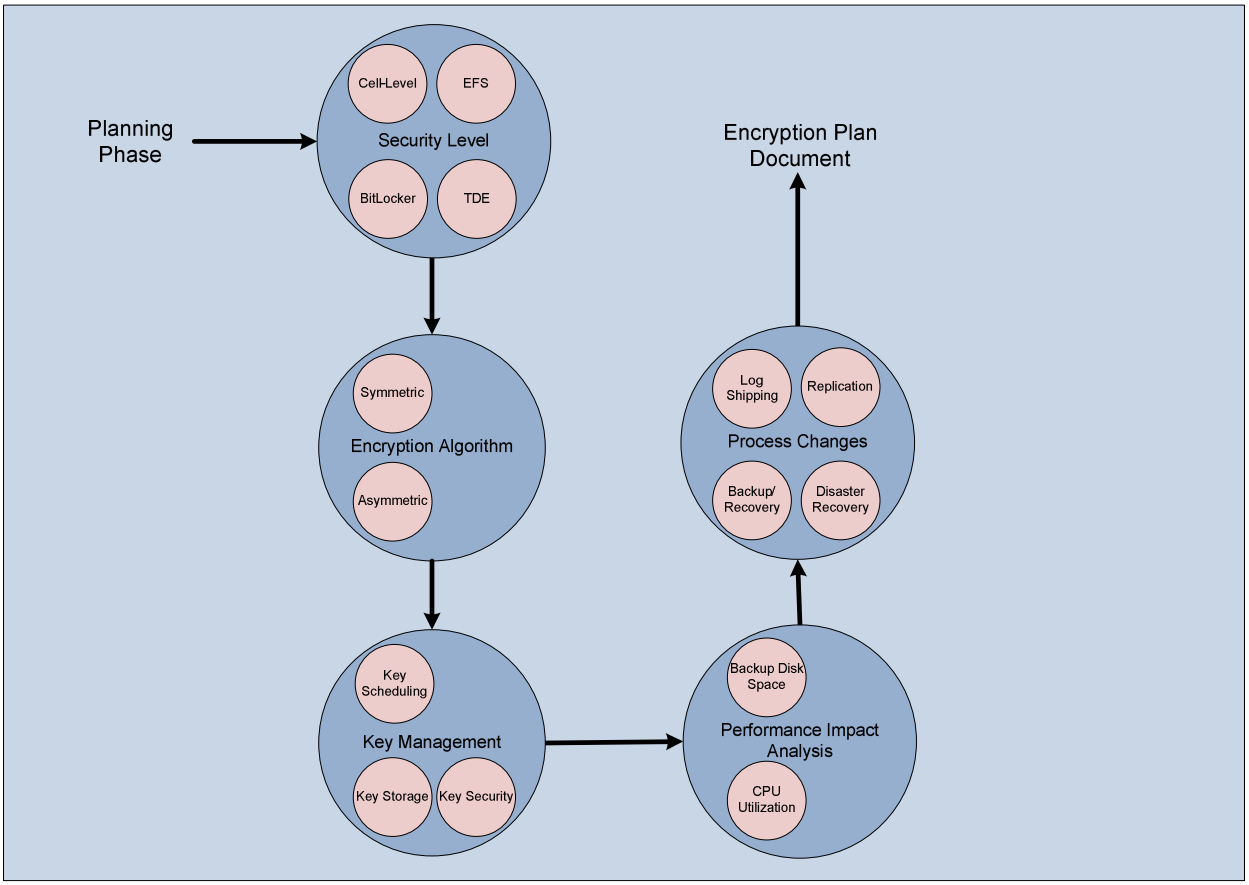


Figure 4 Encryption Planning Process

## Summary

A comprehensive IT security policy provides a layered defense against threats to the system. However, even the most thorough perimeter network and physical defenses do not obviate the vulnerability of plaintext data stored in databases. Data encryption provides a means to protect sensitive data from unauthorized access as a part of a coordinated IT security policy that includes network security, robust authentication and authorization, as well as other physical security considerations. SQL Server and Windows® provide several mechanisms for the protection of data either at the file, database, or data levels.

Transparent database encryption (TDE) is a new technology available in SQL Server 2008 Enterprise Edition which provides a simplified the data encryption option. TDE is a database-level encryption mechanism that reduces the implementation complexity by negating the need to modify the data and / or the client applications. However, the benefits of performance and simplicity are balanced by TDE's potential for data leakage; therefore, for the most sensitive data TDE alone may not suffice as a data security strategy.

Any data protection strategy must weigh the costs and benefits of implementation to arrive at a usable solution that meets the security requirements defined by the business. TDE's protection of sensitive data in low to moderate threat environments may be sufficient for some business requirements while highly sensitive data or data in high threat environments will require the combination of TDE with other encryption mechanisms such as cell-level encryption, EFS, or BitLocker.

## About the Author

Ron is a Senior DBA (MCDBA) who specializes in performance optimization, replication, and security.

# Glossary of Security / Encryption Terms

| | |
|---|---|
| Authorization – | The official management decision given by a senior agency official to authorize operation of an information system and to explicitly accept the risk to agency operations (including mission, functions, image, or reputation), agency assets, or individuals, based on the implementation of an agreed-upon set of security controls.<br>SOURCE: SP 800-37 |
| Authenticate – | To confirm the identity of an entity when that identity is presented.<br>SOURCE: SP 800-32 |
| Authentication – | Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.<br>SOURCE: SP 800-53; FIPS 200 |
| Authentication – | The process of establishing confidence of authenticity.<br>SOURCE: FIPS 201 |
| Authentication – | Encompasses identity verification, message origin authentication, and message content authentication.<br>SOURCE: FIPS 190 |
| Authentication – | A process that establishes the origin of information or determines an entity's identity.<br>SOURCE: SP 800-21 [2nd Ed] |
| FIPS - | Federal Information Processing Standard |
| IT Security Goal – | The five security goals are confidentiality, availability, integrity, accountability, and assurance.<br>SOURCE: SP 800-27A |
| Information Security – | The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.<br>SOURCE: SP 800-53; FIPS 200; FIPS 199; 44 U.S.C., Sec. 3542 |

| | |
|---|---|
| Information Security – | Protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide—<br>1) integrity, which means guarding against improper information modification or destruction, and includes ensuring information nonrepudiation and authenticity;<br>2) confidentiality, which means preserving authorized restrictions on access and disclosure, including means for protecting personal privacy and proprietary information; and<br>3) availability, which means ensuring timely and reliable access to and use of information.<br>SOURCE: SP 800-66; 44 U.S.C., Sec 3541 |
| Key – | A value used to control cryptographic operations, such as decryption, encryption, signature generation or signature verification.<br>SOURCE: SP 800-63 |
| Key Establishment – | The process by which cryptographic keys are securely distributed among cryptographic modules using manual transport methods (e.g., key loaders), automated methods (e.g., key transport and/or key agreement protocols), or a combination of automated and manual methods (consists of key transport plus key agreement).<br>SOURCE: FIPS 140-2 |
| Key Management – | The activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and zeroization.<br>SOURCE: FIPS 140-2 |
| Principal – | An entity whose identity can be authenticated.<br>SOURCE: FIPS 196 |
| Private Key – | A cryptographic key used with a public key cryptographic algorithm, which is uniquely associated with an entity, and not made public; it is used to generate a digital signature; this key is mathematically linked with a corresponding public key.<br>SOURCE: FIPS 196 |

Private Key – A cryptographic key, used with a public key cryptographic algorithm, that is uniquely associated with an entity and is not made public.
SOURCE: FIPS 140-2

Private Key – The secret part of an asymmetric key pair that is typically used to digitally sign or decrypt data.
SOURCE: SP 800-63

Private Key – A cryptographic key, used with a public key cryptographic algorithm, that is uniquely associated with an entity and is not made public. In an asymmetric (public) cryptosystem, the private key is associated with a public key. Depending on the algorithm, the private key may be used to—
1) Compute the corresponding public key,
2) Compute a digital signature that may be verified by the corresponding public key,
3) Decrypt data that was encrypted by the corresponding public key, or
4) Compute a piece of common shared data, together with other information.
SOURCE: SP 800-57

Privileged Accounts – Individuals who have access to set "access rights" for users on a given system. Sometimes referred to as system or network administrative accounts.
SOURCE: SP 800-12

Public Key – A cryptographic key that is used with a public key cryptographic algorithm. The public key is uniquely associated with an entity and may be made public. In an asymmetric (public) cryptosystem, the public key is associated with a private key. The public key may be known by anyone and, depending on the algorithm, may be used to—
1) Verify a digital signature that is signed by the corresponding private key,
2) Encrypt data that can be decrypted by the

| | |
|---|---|
| | corresponding private key, or<br>3) Compute a piece of shared data.<br>SOURCE: SP 800-57 |
| Public Key – | A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and which may be made public; it is used to verify a digital signature; this key is mathematically linked with a corresponding private key.<br>SOURCE: FIPS 196 |
| Public Key – | A cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and that may be made public.<br>SOURCE: FIPS 140-2 |
| Public Key Certificate – | A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a subscriber to a public key. The certificate indicates that the subscriber identified in the certificate has sole control and access to the private key.<br>SOURCE: SP 800-63 |
| Public Key Certificate – | A set of data that unambiguously identifies an entity, contains the entity's public key, and is digitally signed by a trusted third party (certification authority).<br>SOURCE: FIPS 196 |
| Public Key Certificate – | A set of data that uniquely identifies an entity, contains the entity's public key, and is digitally signed by a trusted party, thereby binding the public key to the entity.<br>SOURCE: FIPS 140-2 |
| Public Key (Asymmetric) Cryptographic Algorithm – | Public key cryptography uses "key pairs," a public key and a mathematically related private key. Given the public key, it is infeasible to find the private key. The private key is kept secret while the public key may be shared with others. A message encrypted with the public key can only be decrypted with the private key. A message can be digitally signed with the private key, and anyone can verify the signature with the public key.<br>SOURCE: SP 800-46 |
| Public Key Infrastructure – (PKI) | A set of policies, processes, server platforms, software and workstations used for the purpose of administering certificates and public-private key |

| | |
|---|---|
| | pairs, including the ability to issue, maintain, and revoke public key certificates.<br>SOURCE: SP 800-32 |
| Public Key Infrastructure – | An architecture which is used to bind public keys to entities, enable other entities to verify public key bindings, revoke such bindings, and provide other services critical to managing public keys.<br>SOURCE: FIPS 196 |
| Secret Key – | A cryptographic key that is used with a secret key (symmetric) cryptographic algorithm, that is uniquely associated with one or more entities and is not be made public. The use of the term "secret" in this context does not imply a classification level, but rather implies the need to protect the key from disclosure.<br>SOURCE: SP 800-57 |
| Secret Key – | A cryptographic key that must be protected from unauthorized disclosure to protect data encrypted with the key.<br>SOURCE: FIPS 201 |
| Secret Key – | A cryptographic key that is uniquely associated with one or more entities. The use of the term "secret" in this context does not imply a classification level, but rather implies the need to protect the key from disclosure.<br>SOURCE: FIPS 198 |
| Secret Key – | A cryptographic key, used with a secret key cryptographic algorithm, that is uniquely associated with one or more entities and should not be made public.<br>SOURCE: FIPS 140-2 |
| Secret Key (symmetric) Cryptographic Algorithm – | A cryptographic algorithm that uses a single secret key for both encryption and decryption.<br>SOURCE: FIPS 140-2 |
| Secret (Symmetric) Key Encryption – | This is the traditional method used for encryption. The same key is used for both encryption and decryption. Only the party or parties that exchange secret messages know the secret key. The biggest problem with symmetric key encryption is securely distributing the keys. Public key techniques are now often used to distribute the symmetric keys.<br>SOURCE: SP 800-46 |

| | |
|---|---|
| Security Requirements – | Requirements levied on an information system that are derived from laws, executive orders, directives, policies, instructions, regulations, or organizational (mission) needs to ensure the confidentiality, integrity, and availability of the information being processed, stored, or transmitted.<br>SOURCE: SP 800-53 |
| Security Requirements – | Requirements levied on an information system that are derived from applicable laws, Executive Orders, directives, policies, standards, instructions, regulations, or procedures, or organizational mission/business case needs to ensure the confidentiality, integrity, and availability of the information being processed, stored, or transmitted.<br>SOURCE: FIPS 200 |
| Symmetric Encryption Algorithm – | Encryption algorithms using the same secret key for encryption and decryption.<br>SOURCE: SP 800-49 |
| Symmetric Key – | A cryptographic key that is used to perform both the cryptographic operation and its inverse, for example to encrypt and decrypt, or create a message authentication code and to verify the code.<br>SOURCE: SP 800-63 |
| Symmetric Key – | A single cryptographic key that is used with a secret (symmetric) key algorithm.<br>SOURCE: SP 800-21 [2nd Ed] |
| Verification – | The process of affirming that a claimed identity is correct by comparing the offered claims of identity with previously proven information stored in the identity card or PIV system. See Identity Verification.<br>SOURCE: FIPS 201 |