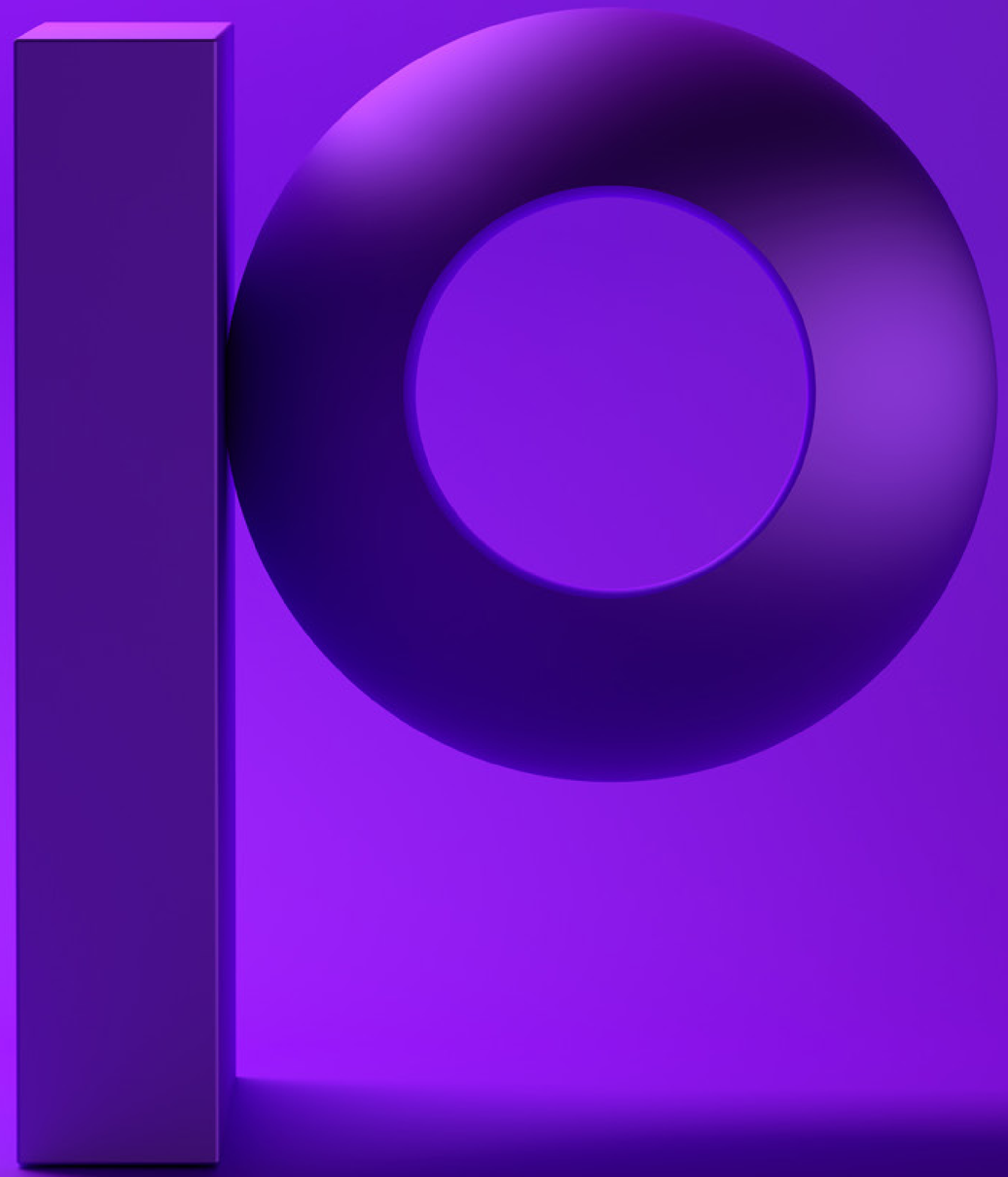
precisely

IBM i Encryption 101



Introduction

For more than 30 years, IBM i has been used by organizations in every industry. Known as a transaction processing powerhouse, IBM i systems are relied upon by organizations focused on retail, entertainment, manufacturing, financial services, and more. Everywhere they are used, IBM i servers are trusted to run businesses' most critical workloads and hold their most sensitive data.

While corporate insiders have long been aware that the data on an IBM i could be used for personal profit, the IBM i has only more recently become known as a source of high-value data by outside intruders. Just one instance of a data breach can cost an enterprise hundreds of thousands of dollars in remediation fees, regulatory fines, brand damage, legal fees, and more. As cybercrime increases in sophistication and frequency, the costs and risks of sensitive data exposure will only increase.

Encryption, a form of cryptography, is an essential security measure for IBM i systems that adds a layer of protection against data breaches and exposure. Cryptography is a foundation of sound IT security that protects the business, and encrypting data is mandated by regulations such as HIPAA, Sarbanes-Oxley, PCI DSS, and many others.

How Encryption Works

Data encryption algorithmically changes data from its original format into a new, unreadable form called ciphertext. For example, when encrypted, the name “Susie Smith” might appear as “K5d&*p2Mg^Z”. You must have access to the proper encryption key and the encryption algorithm used to encrypt the data to decrypt it and make it readable again. This is true whether data is at-rest (stored in a static location) or in-motion (being transferred over the network).

Encryption Algorithms

An encryption algorithm performs the process by which data is converted into ciphertext and then restores it to its readable form for the user who has the correct encryption key. Encryption algorithms are public information, and attempts are routinely made to break them to ensure their continued security.

The US National Institute of Standards and Technology (NIST), is a globally-respected organization that sets standards for data encryption and cryptography algorithms. As technology changes, so do NIST recommendations. Therefore, algorithms that were once the gold standard for cryptography become obsolete once they are broken.

Currently AES, or Advanced Encryption Standard, is the NIST-recommended encryption algorithm for providing security against data breaches. Implemented in 2002 as the standard for the federal government, AES uses longer key lengths and enables faster encryption than its predecessor encryption algorithms.

For a current list of algorithms and their approval status for encryption and decryption, please see NIST Special Publication 800-131Ar2.

Encryption Keys

To encrypt data, an encryption key must be created that will be used by the encryption algorithm to transform the data into ciphertext. A key can be a number, a word, or a string of random characters. Keys can also be of different lengths (e.g. 128, 192 or 256 bits). As long as both sender and recipient know the key, they can use it to encrypt and decrypt data.

Each encryption key can be unique. Because algorithms are public information, **encryption keys must be kept secret**. Encryption keys are similar to house keys. Many homes have similar types of locks, but each has its own unique key used to lock and unlock the door. If an intruder steals your encryption key, it's just like stealing the key to your house. **Hackers don't break encryption algorithms; they find the keys**.

Symmetric and Asymmetric Encryption

There are two basic techniques for encrypting data: symmetric encryption (also called secret key encryption) and asymmetric encryption (also called public key encryption).

Symmetric encryption

Symmetric encryption is the oldest and best-known encryption technique. As the name implies, symmetric cryptography uses the same key to encrypt and decrypt data.

Symmetric encryption is frequently used for encrypting valuable data, such as credit card information. When a credit card number is encrypted, the ciphertext version is a string of random characters that no longer looks like a valid credit card number. When the string is decrypted using the same encryption key, the data is transformed back to its original state – a valid credit card number. AES is a symmetric key algorithm.

Asymmetric encryption

Asymmetric encryption requires two keys: a public key and a private key. This type of encryption uses one key to encrypt data and the second key to decrypt it.

In asymmetric encryption, the private key is kept secret, and the public key is shared with the party that is meant to decrypt the information. Data can also be encrypted with a public key and decrypted with the private key.

One of the primary uses of asymmetric encryption is authentication. After a user exchanges a public key with trading partners, the corresponding private key can be used to digitally sign data, allowing the decryptor to verify the authenticity of the sender.

Encryption Key Management

Since encryption keys must remain secret, sound methods of creating, distributing, and storing encryption keys are essential. For example, encryption keys should not be stored on the same system as the sensitive data they protect. Third-party key management tools are offered to manage creation, activation, use, rotation, expiration, retirement, and destruction of encryption keys. Also crucial in key management are separation of duties, dual-control processes, and more.

Key rotation

Periodically changing the encryption keys (sometimes called “key rotation” or “key rollover”) is vital to the overall security of your protected data. Data encryption keys should be changed at appropriate intervals.

The appropriate interval for changing keys depends on several variables, including the amount of data the key protects and the sensitivity of that data. This interval is called the cryptoperiod of the key and is defined by NIST in Special Publication 800-57 “Key Management Best Practices.

Separation of duties

The concept of separation of duties in encryption key management requires the administrators responsible for key management to not have access to the sensitive data that will be encrypted, and database administrators should not have access to the encryption keys.

Dual control

Sound key management practices also require dual control, meaning that more than one security administrator must authenticate before accessing and managing keys.

Masking

Many compliance regulations such as PCI DSS and HIPAA require that data be completely or partially hidden from users who are authorized to decrypt data but do not need to see it in its entirety to do their jobs.

The process of partially obscuring encrypted data when it is decrypted and displayed to a user is called masking. The most fundamental difference between encryption and masking is that encryption transforms the original data into ciphertext; whereas masking does not transform the data, it only obscures data from view.

Data masking is an important security feature that is often implemented along with encryption. For example, if you encrypt a bank account number, bank tellers may only need to see the last 4 digits of that number to verify the account with a customer (for example, use masking to show ********1234), but a private investment advisor may need to see the full account number to help customers manage and transfer funds.

Hashing

Hashing is a cryptography function that permanently converts clear text to an unreadable fixed-string hash code. Hashing has a variety of uses, including authentication and establishing message integrity. For example, an email might be hashed to a string such as 5f4dcc3b5aa765d61d8327deb882cf99. By hashing the email text again on receipt, if the same hash value is produced, you can verify that the email hasn't changed.

Hashing can also be used for authentication purposes by an operating system or application. For example, a hashing algorithm takes a user's password and ID and hashes them together into an unrecognizable string. Then the OS or application stores the hashes, not the actual clear text passwords, in a secure location.

When a user logs in, the algorithm runs the hash of the userid and entered password and compares it with the hash that was stored. If the user changes his or her password, the OS or application would perform the hash once again using the userid and new password and store this new hash away for future authentication purposes.

For more information, click here visit the NIST Computer Security Resource Center.

Best Practices for Encrypting Data on IBM i

Take advantage of IBM i field procedures

IBM i Field Procedure, or FieldProc, support was developed in operating system release 7.1 to allow data in Db2 to be easily encrypted. The most significant advantage of field procedures is that you do not have to rewrite application programs to include the encryption and decryption API calls to encrypt or decrypt data. Field Procedure support allows a user to associate a field procedure program with a field or column of data in a Db2 file or table.

When an OS interface or an application reads or writes data from the Db2 file, the Field Procedure program is called to either encrypt or decrypt, depending on the read or add/update Db2 operation being performed. The Field Procedure program also handles encryption key management support.

The Field Procedure program can be written by customers or obtained from a vendor, such as Precisely. The field procedure program handles all encryption or decryption implementation and interfaces with key management solutions. It essential that you use a state-of-the-art, NIST-Certified encryption algorithm such as AES when implementing your field procedure support.

IBM i full disk and backup tape encryption

Full disk encryption and backup tape encryption helps your organization secure sensitive data. Backup tape encryption protects data while it is on physical tape media, making the tape useless to someone who does not have the encryption key required to restore from the encrypted tape. Full disk encryption protects data stored on a disk drive but does NOT provide security for a system that is up and running.

For example, if someone were to obtain your encrypted disk drive, they could not read the data unless they have the encryption key used to encrypt it. However, if the system is up and running in your production environment, full disk encryption provides no additional security. All data read from the disk will be automatically decrypted during the read of the data. Once the read is complete, all data read from the disk drive is decrypted and will be presented as clear text to the OS or application reading the data. If your security requirement is to protect data via encryption and to prevent users from seeing clear text, you must use application-level encryption using Db2 Field Procedures or encryption calls from the application itself.

As part of the operating system, IBM i has an encryption software library that can be used to perform disk-level encryption of an auxiliary storage pool of disk drives and encryption of your system backups. This support is available as OS options 44 and 45. Because the encryption is done in software when using these OS options, system performance will be affected.

The optimal solution for disk-level encryption and tape encryption is available in external storage (SAN) using the hardware option for full disk encryption and in IBM tape hardware support. Full disk encryption and tape encryption gives you peace of mind if you ever have to dispose of a failed disk drive or misplace a tape containing your sensitive data. The hardware support for full disk encryption or tape encryption does not impact system performance.

Keep encryption keys secure

If a malicious actor gains access to your encryption keys, they can decrypt sensitive information that was previously protected. This can have severe consequences, including data breaches, identity theft, financial losses and legal consequences. To prevent these consequences, it is crucial to implement robust security measures to protect encryption keys.

Separation of duties

Ensure IBM i security administrators such as QSECOFR and any user with All Object (*ALLOBJ) does not have access to data encryption keys.

Dual control

More than one security administrator should authenticate before accessing and managing keys. All access to, and use and management of, encryption keys should be logged in IBM i audit journals.

Backup and availability

Encryption keys should be backed up in real time to match your organization's standards for system availability.

Encryption key rotation

Most IBM i customers should rotate data encryption keys at least once a year.

Caching

Encryption keys are used frequently when batch operations are performed on sensitive data. Therefore, it's not unusual for a batch program to perform millions or tens of millions of encryption and decryption operations. While retrieving an encryption key from the key server may be efficient, performance may suffer when keys need to be retrieved many times. This can be addressed by caching encryption keys in the local environment.

Secure key caching should be performed in separate program modules, such as a service program, and should not be cached in user programs where they are more subject to discovery and loss. Any module caching an encryption key should have debugging options disabled and visibility removed. Secure key caching is critical for system performance, and care should be taken to protect storage.

Mask to fit your business needs

Data masking isn't just a good idea; compliance regulations such as PCI DSS require data masking. Ensure your encryption solution provides the masking options your business requires, whether that's masking all but the last four characters, the first five characters, or other custom masking methods.

Assess encryption performance

The performance of your encryption solution is of utmost importance. For data encryption to be effective, it must make your data unreadable to unauthorized users and decrypt it for authorized users, without impacting system performance. If encryption takes too long or consumes too many resources, consider switching to a different algorithm, or tuning the settings for your data encryption product.

Precisely Assure Encryption

While encrypting sensitive data on IBM i systems used to be difficult to implement due to the need to rewrite applications, today's technology is a vast improvement over older IBM i encryption support. Modern encryption methods, such as those leveraged by Assure Security product, offer complex encryption and key management technology.

Assure Security's encryption capability, **Assure Encryption, is the only NIST-certified AES encryption solution for IBM i**. It provides encryption libraries that integrate with IBM i Field Procedures to automatically encrypt and decrypt data in Db2 columns without requiring application changes. Commands are also available to encrypt backup tapes, Save files, IFS files and more. In addition, a complete set of APIs is provided to secure data in individual database fields from IBM i OPM or ILE applications built with RPG, COBOL, and other languages.

Performance-optimized to encrypt and decrypt IBM i Db2 data without impacting application performance, Assure Encryption also offers built-in masking and access auditing.

To ensure encryption keys remain secret, Assure Encryption seamlessly integrates with Townsend Security's Alliance Key Manager, a FIPS 140-2 compliant key management solution, as well as other OASIS KMIP-compliant key managers, for comprehensive key management.

Your customers, business partners, and employees trust you to protect confidential information from unauthorized access and theft, and regulations mandate that you provide privacy for personally identifiable information, cardholder information, healthcare information, and more. Stiff fines and negative publicity await those who do not properly protect data from breach. Get started protecting the data on your IBM i today with Assure Encryption.

precisely

As a global leader in data integrity, Precisely ensures that your data is accurate, consistent, and contextual. Our portfolio, including the Precisely Data Integrity Suite, helps integrate your data, improve data quality, govern data usage, geocode and analyze location data, and enrich it with complementary datasets for confident business decisions. Over 12,000 organizations in more than 100 countries, including 93 of the Fortune 100, trust Precisely software, data, and strategy services to power AI, automation, and analytics initiatives. Learn more at www.precisely.com.

www.precisely.com