Introduction to Application Encryption

TABLE OF CONTENTS

APPLICATION ENCRYPTION OVERVIEW

Proactively encrypting sensitive data as soon as possible is one of the best approaches to enterprise data security. By implementing encryption up front at the application layer, exposure of sensitive clear-text data is minimized. This "top-down" approach to encryption reduces the need for secondary encryption platforms like full-disk hard drive encryption software, which are often cumbersome to deploy and manage and often lack important security certifications. For this reason, many organizations are seeking ways to incorporate hardened encryption into client-facing applications. Futurex accommodates this need with its application encryption technology.

Application encryption is full-service cryptographic functionality available through the Key Management Enterprise Server (KMES) Series 3 that incorporates general-purpose data encryption and key management technology into applications. Application encryption allows organizations to encrypt entire files or specific fields of data at the application level, before it is stored. This whitepaper is focused on Futurex's application encryption functionality as well as important best practices for any organization considering deploying it.

SECURITY

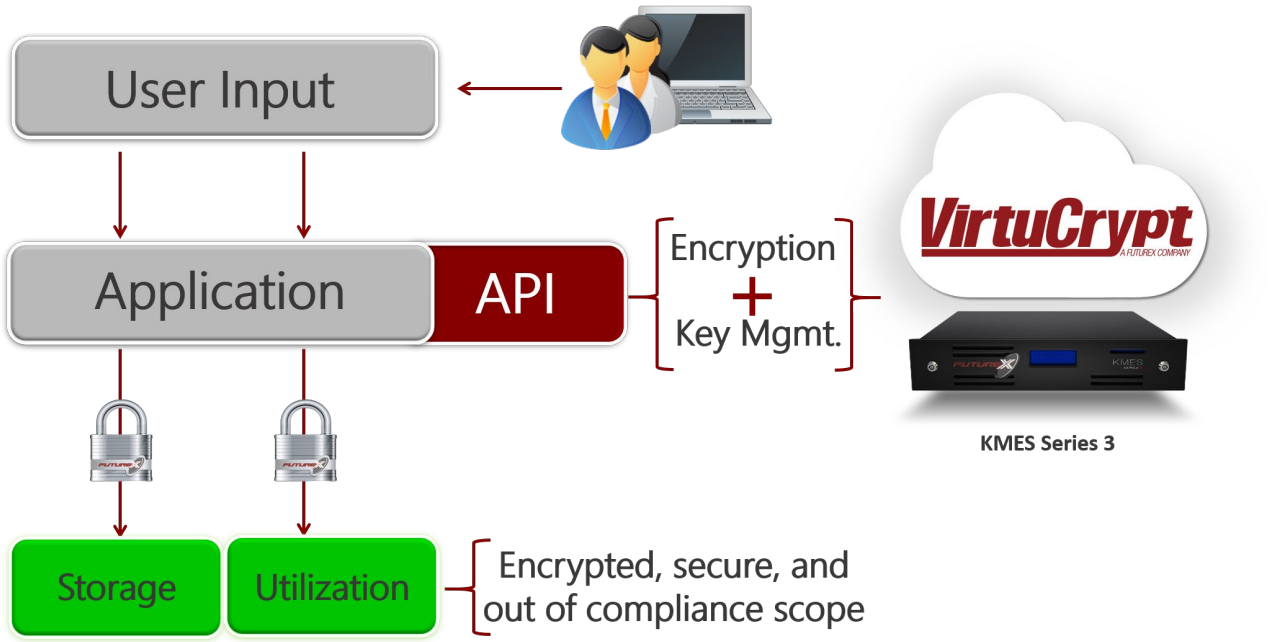
Application encryption technology brings the cutting-edge cryptographic power of Futurex's Hardened Enterprise Security Platform directly into an organization's own applications, or those from a compatible third party. This is a sophisticated and secure option for protecting sensitive data. With application encryption, the data is encrypted immediately upon ingestion into the application, which protects it across its entire lifecycle, from input to storage. Application encryption can work together with other cryptographic techniques and use cases such as Point-to-Point Encryption (P2PE), tokenization, or any other supplementary encryption-based data security mechanism.

Futurex's primary application encryption platform is the KMES Series 3. The KMES is a complete key management and general-purpose encryption solution, equipped with an internal Secure Cryptographic Device (SCD) for key storage and cryptographic processing. It is fully compliant with FIPS 140-2 Level 3, PCI HSM, and all other major industry standards for security.

COMPLIANCE

Enterprise organizations must process data in accordance with certain standards of security compliance relevant to each industry. Two common examples are found in the financial payments and healthcare industries. Financial institutions and merchants must manage and protect their

customer's payment data in a manner that is compliant with Payment Card Industry Data Security Standard (PCI DSS) requirements. Similarly, healthcare organizations must adhere to the Health Insurance Portability and Accountability Act (HIPAA) when processing patient data. Maintaining compliance requires ongoing cost and maintenance. However, organizations who practice encryption can alleviate much of this expenditure. PCI DSS, HIPAA, and most similar data regulations apply to clear-text data. Once data is encrypted, it is unreadable and typically not considered "in scope" for compliance requirements. As such, application encryption technology often rapidly achieves return on investment for organizations who operate under compliance mandates, coming in the form of both time and money saved on compliance audits and maintenance.



Application Encryption Overview

ADAPTABILITY

Many companies balk at application encryption out of fears that processes required to secure their stored data will affect system performance. Futurex's robust API allows organizations to fine tune their application encryption by only targeting specific files, data types, or columns of data that are deemed sensitive. This reduces unnecessary encryption of non-sensitive data and minimizes any effects on system performance. The API supports a high amount of customization that clients can tailor to fit their data streams.

The flexibility of the KMES Series 3 allows individual customers to choose how automated, or how much user interaction is required, which is typically predefined by the customer's security policy. The KMES Series 3 can be fully automated after initial setup and loading of the major keys. Application

encryption requires integration by incorporating the KMES Series 3's application programming interface (API) into the endpoint application. This will be described in greater detail later in this document. The integration application can be written in any language that allows for basic TCP/IP support (Java, C, C++, etc.), or using a web services (RESTful) API.

GLOBAL MANAGEMENT USING THE HARDENED ENTERPRISE SECURITY PLATFORM

What sets Futurex's Application Encryption apart from others in the marketplace is the Hardened Enterprise Security Platform, Futurex's complete product line of HSMs, key management solutions, cryptographic management platforms, and cloud-based services. These devices are built on Futurex's Base Architecture Model, a common code base and shared API, that ensures all devices are fully interoperable, scalable, and easily expanded over time.

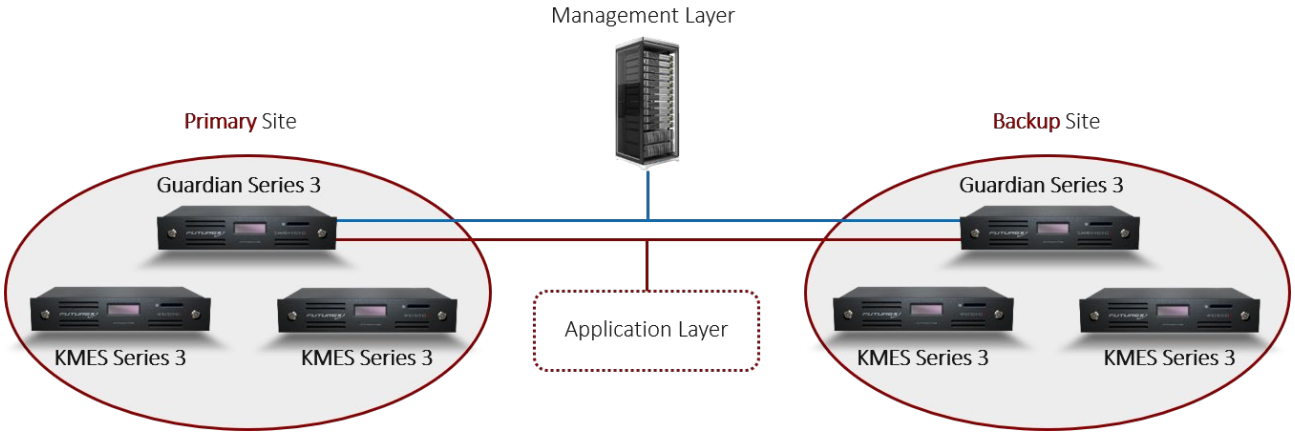
Hardened Enterprise Security Platform Structure

Futurex's Hardened Enterprise Security Platform also features centralized management platforms that give administrators the ability to manage their cryptographic infrastructure from a unified platform. The Futurex Guardian Series 3 vastly simplifies infrastructure management by allowing administrators to control multiple geographically-separated devices from a single platform and removes the need to physically access each device. The Guardian provides a consolidated interface for system monitoring, disaster recovery, redundancy, high availability, automated administration, and real-time alerts.

DEVICE CLUSTERING

Many organizations manage large data ecosystems that consist of multiple data centers and

cryptographic sites supporting multiple applications. In these instances, multiple KMES devices may need to be active in multiple locations. The Guardian supports device clustering, which gives administrators the option to organize devices into functional groups with the redundancy necessary for fault tolerance. If one unit in the group should ever fail, the other units would automatically take up the transaction load. For example, in the event of a disaster (natural or otherwise), having a load-balanced KMES Series 3 device in a second remote location means the Guardian would re-route traffic to the off-site device to compensate.



Clustering Example

PEERING

Masterless Peering is a method of sharing data between Futurex devices spread across multiple, geographically dispersed locations. Specifically, this feature has been implemented to share certificates, PKI pairs, certificate authorities, logs, and users. Using this method, data is replicated across multiple devices, eliminating a single point of failure where one device is charged with the data integrity of all devices in a group.

Peering is achieved by assigning every object in a KMES Series a unique OID (object identifier) represented by a 64-bit integer and a "Last modified" date to be stored in the database. The OID allows objects or keys to be commonly referenced across multiple peered devices. The Peers tab in the KMES Series 3 management application is used to manage these operations. Devices displayed in the Peers tab are organized into the following device role categories:

- Primary Device: The configuration details on this device will automatically be replicated to any additional devices added to the device group. The primary device also functions in the same role as a production device.
- Temporary Primary Device: Designating a device as a temporary primary device will allow that device to take over primary functionality if the primary device becomes unavailable. The temporary primary device will be automatically designated as a production device and the

original primary device will be restored as soon as its issues are resoWHolved and it is recognized and authenticated by the Guardian.

- **Production Device:** Production devices will begin actively processing transactions as soon as they have been synchronized with the group. Multiple production devices may be added to an individual device group.
- **Backup Device:** Designating a device as a backup device will cause it to remain synchronized with the group, but not process transactions, until a production device is removed from service, at which point it will automatically begin processing transactions. The use of backup devices is optional, and multiple backup devices may be added to an individual device group.

REMOTE MANAGEMENT

Futurex offers a high level of remote access functionality with their application encryption platform that is unrivaled by others in the industry. The Guardian 3 is equipped with a web-management portal that allows administrators to bring management of their cryptographic infrastructure with them on the road, at home, or virtually anywhere through a web-enabled device. Through the Guardian Web Portal, users can monitor their cryptographic operations, perform key management, create custom reports in many formats, and view operational information about their Futurex devices.

As an additional, high-security access method, Futurex offers the Excrypt Touch, a portable, touchscreen-based configuration and key management tablet designed to remotely configure and manage the KMES Series 3 and any network-attached Futurex devices. Once connected via a routable network path, a single Excrypt Touch can manage all authenticated Futurex devices, no matter where they are physically located. The simple, easy-to-use, touch screen-based interface aids in the rapid configuration and management of Futurex units, resulting in reduced training and operational time.

Guardian Series 3 Web Management Portal and Excrypt Touch Tablet

KEY MANAGEMENT METHODOLOGIES

Application encryption requires key management implementation in conjunction with general purpose encryption. Furthermore, it can require especially intensive key management functionality in large infrastructures where multiple applications take part in the encryption process. Adding to the importance of key management, application encryption is essentially "cradle to the grave" encryption which requires long-term key storage. Key management is important in any infrastructure, but application encryption raises the stakes in this aspect. In order to effectively implement application encryption, a high-volume key management solution with a robust API is a must.

A HARDENED KEY MANAGEMENT SOLUTION

Many organizations lower the bar for security in attempt to meet the key management requirements of application encryption. A common mistake is performing key management within the application itself. This is troublesome for multiple reasons, but the most important is vulnerability. This model places the keys within the application, which subjects them to the same network threats as the application itself. It also requires the extension of key management access to each application manager, which raises the risk for internal theft or negligence.

Another common mistake with key management for application encryption is using software-based key storage programs. Software-based encryption programs are inherently flawed due to their vulnerability to malware, keylogging, and other attacks that attempt to determine encryption keys.

This is why Futurex uses real-world hardware security modules to store all encryption keys. Hardware-based key management solutions are a far more secure option; however, many security providers lack a solution scalable and flexible enough for enterprise-level application encryption. Futurex's KMES Series is a robust, centralized key management server. It stores all keys in an internal secure cryptographic device that is fully compliant with FIPS 140-2 Level 3 requirements, which ensures a sophisticated level of physical security, including:

- Tamper-responsive circuitry that erases sensitive data upon detection of any intrusion attempt
- Physical security barriers that prevent access to internal components
- Digital signatures of cryptographic modules that prevent substitution attacks

In addition to security, Futurex KMES Series 3 also provides one of the most robust and functional key management platforms available. They KMES vastly simplifies key management for application encryption by performing key management and general-purpose encryption from a single, unified interface. For infrastructures with multiple applications with integrated encryption, the KMES provides a centralized platform from which to distribute to key and ties them back into the internal HSM for secure, long-term storage.

KEY GROUPS/CLASSES

With centralization of keys comes centralization of access. With the KMES' customizable user management system, key management access can be restricted to just a few security administrators with multi-factor authentication mechanisms. Furthermore, keys can be organized into groups, each with their own specific user privilege configurations, which only grants certain administrators access to keys specific to their needs, thus, maintaining the security principle of least privilege to minimize unnecessary risk.

Key grouping also comes in handy in multi-application infrastructures. Keys can be grouped and distributed based on their destination application. This is useful if certain applications have different requirement from a key management perspective.

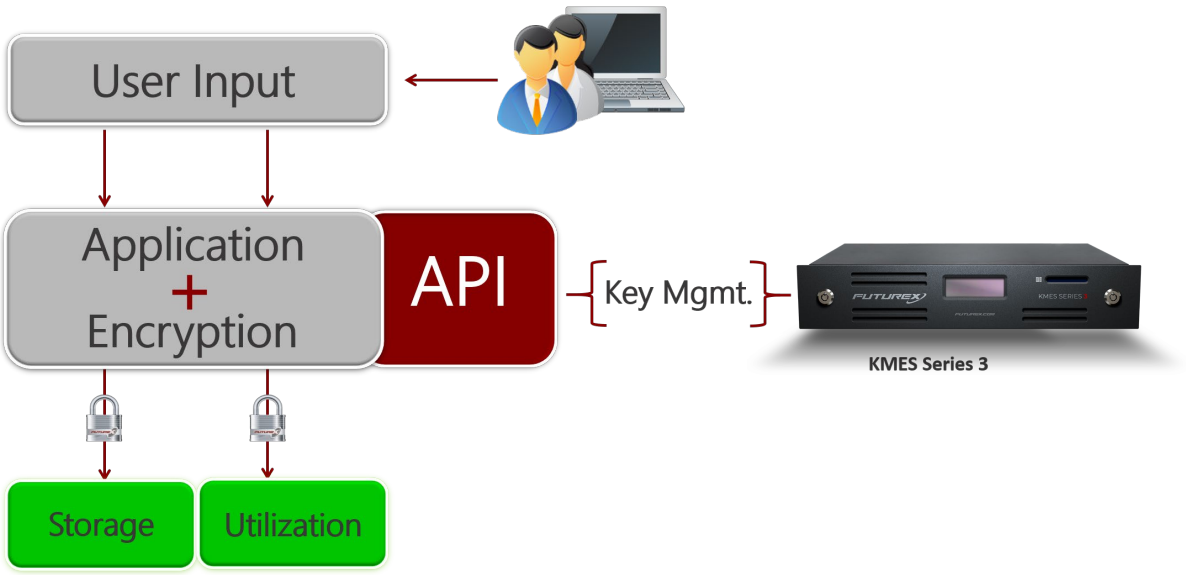
KEY ROTATION

Administrators can also choose the implementation algorithm for when those keys are distributed. Administrators can set up automated key rotation by instructing the KMES to retrieve keys in order of their expiration date, a custom order, or they can keep the process completely manual if desired. The flexibility of the KMES Series 3 allows individual applications to choose how automated, or how much user interaction is required

KEY MANAGEMENT FOR CLIENT-SIDE ENCRYPTION

Depending on the throughput requirements of the environment, some organizations may wish to utilize application encryption to support client-side encryption. Client-side encryption is an enterprise encryption model that encrypts the data within the application itself, without sending the raw data to the processing HSM. Using this method, the KMES Series is only used for key management.

With the client layer handling the encryption process, key management is still handled by the KMES Series 3 functionality integrated into the application layer. This model is beneficial for organizations with exceptionally high throughput requirements. With the processing resources required for the encryption process being provided by the application, it allows the KMES to focus solely on key management. This technique is conducive to faster, high-volume encryption, without the need to purchase additional hardware.



Client-Side Encryption Overview

PERMISSIONS

An important security feature of the KMES Series is its robust set of permission settings, which allow administrators to configure high levels of customizations for both users, keys, and various other cryptographic objects. This allows administrators to dictate precisely which users can access certain keys or objects.

USERS

For managing users, the KMES Series supports creating of multiple user groups, each of which can be configured with their own permissions for accessing keys, objects, and other functions. Some of the specific parameters include:

- Database backup and restore
- Manage and print reports
- Manage encryption device groups
- Manage hosts/networks
- Manage certificates
- Manage templates
- Manage users
- Manage keys
- Update system configuration
- View logs
- View peers

KEYS AND OTHER CRYPTOGRAPHIC OBJECTS

In addition to managing access on user side, permissions specific to keys and object may also be managed. For example, granting a user access to a certain key group is a two-part process. First, the user must be granted the Manage Key permission covered in the previous section. Second, that particular user group must be given access to specific keys in the key group settings. By default, the owner of the key group and the admin group will have full rights to the object and all other user groups will have none.

Permissions must be set for the following objects:

- Reports
- Encryption Device Groups
- Users
- System Configuration
- Logs
- Symmetric Keys
- Certificate Authorities
- Certificates
- Hosts/Networks
- Mailer/Z-Fold Printing Templates

COMMUNICATION METHODS

A common obstacle for many organizations wishing to implement application encryption is a general fear of making significant changes to their application code. While a certain amount of modification is unavoidable, impacts on applications can be minimized by implementing an encryption solution with an application programming interface (API) that can easily integrate into existing application code. For this reason, Futurex's KMES Series 3 uses a robust API, bridging the gap and allowing interaction between the client applications and the functionality of the KMES in a number of clientless and client-based API languages.

CLIENT-BASED APIS

PKCS #11

Many organizations choose to implement standardized APIs in order to provide rapid integration with compatible HSMs. PKCS #11 is a common choice for software vendors who utilize encryption in their applications. PKCS #11 (Public-Key Cryptographic Standard #11) is a standardized API in the C programming language that allows easy automation of cryptographic operations such as encryption,

decryption, signing, and verifying. PKCS #11 also provides key management functions such as key generation, derivation, and importing. All of the most common cryptographic ciphers are supported by the library, including 3DES, AES, and RSA.

The KMES Series 3 supports PKCS #11, which provides a software library that links the PKCS #11 API defined in the standard with proprietary applications to perform the cryptographic operations. Clients who use PKCS #11 in their applications can integrate application encryption with the KMES Series 3 with minimal downtime and system impact.

Java

Java's security API is another common set of APIs and implementation tools for encryption-based data security. The Java library is fully supported by the KMES Series 3, which allows the device to bolster client applications with a range of encryption mechanisms, which include: digital signatures, general encryption, message authentication, and key management functionality.

CLIENTLESS APIS

REST APIs

For implementing encryption into web and cloud applications, representational state transfer (REST) APIs can be extremely useful. A REST API uses HTTP functionality to give applications access to the KMES Series 3 on a "stateless" transaction basis, with "stateless" meaning that all commands from the application are independent, and no data is stored in the application between transactions.

REST APIs are commonplace in web applications and cloud providers. They generally have low bandwidth demands and tend to lend themselves better to fast performance and scalability. The increased use of cloud computing and encryption-as-a-service has lead to increased integration of REST APIs into encryption platforms. The KMES Series 3 is no exception and comes complete with a robust REST API ready bring application encryption to any web application.

KMIP

The Key Management Interopability Protocol (KMIP), an OASIS standard, streamlines key management activity between a key management server and endpoint applications and devices. KMIP allows full-lifecycle key management for both asymmetric and symmetric keys for use in application encryption. KMIP solves many problems related to key management for large organizations operating multi-application environments by providing a standardized exchange protocol for distributing encryption keys across multiple clients, servers, and application.

The KMES Series 3, when enabled to work with the KMIP cryptographic library, acts as a KMIP server, facilitating the exchange of cryptographic keys and data encryption. KMIP communications travel

across a secured TLS network, which requires that both the client and the KMES Series 3 have valid certificates and private keys for mutual authentication to occur.

ENCRYPTED DATA STRUCTURES

To aid in adoptability and integration, Futurex uses NIST-standard format-preserving encryption (FPE) using the FF1 algorithm. FPE allows encrypted information to be integrated into existing environments with no database changes by encrypting data in the same format as the original data. Take for example a credit card primary account number (PAN), a common use case for encryption. A PAN is typically between 8 and 19 numeric digits, and when using format-preserving encryption, the encrypted output will have the same number of digits. For organizations with strict database schemas, format-preserving encryption ensures application encryption data be integrated without making major changes to database or alignments.

DATA HEADERS

Futurex's application encryption API allows administrators to customize how their data is encrypted. One aspect of that is data headers. Data headers contain metadata related to each encryption transaction. This includes the amount of data padding used, the key group identifier, ciphers, initialization vectors, etc. Each encrypt or decrypt command has an option allowing users to include data headers in the encrypted output. Some organizations may wish to encrypt them if resources allow or if the metadata in the header is deemed proprietary.

INDEPENDENT METADATA

The same metadata can be left out of the encryption process and exported separately. In most cases, the independent data will not contain inherently sensitive information. Choosing to leave this data unencrypted can greatly reduce the amount of encrypted data stored and free up encryption resources for other transactions. In many cases, organizations will only need to encrypt small pieces of information. If the metadata associated with the encryption is also encrypted, the encrypted footprint for that data can increase many times over. The KMES Series 3 API allows users to customize the metadata exchange and decide if this metadata needs to be encrypted.

CONCLUSION

Application encryption is a sophisticated solution for data protection. Encrypting within the application is one of the most effective ways to ensure data protection, however, it also requires a very adaptable and secure platform capable of performing both general-purpose encryption and key management. While many organizations sacrifice security, both physical and logical, to make their products flexible enough for application encryption, Futurex does not. Futurex application encryption

is backed by hardened cryptographic devices that meet stringent levels of compliance for physical and logical security.

Futurex's application encryption through the KMES Series 3 is a secure, full-service, and highly-available cryptographic solution, capable of meeting the application encryption needs of any enterprise organizations. For more information and to schedule a demonstration of application encryption and the Hardened Enterprise Security Platform, contact Futurex today.

FUTUREX.COM

OFFICE: +1 830 - 980 - 9782 TOLL FREE: 800 - 251 - 5112
864 OLD BOERNE ROAD, BULVERDE, TEXAS, USA 78163

FUTUREX ENGINEERING CAMPUS