THALES

www.thalesesecurity.com

‹Thales eSecurity›

VORMETRIC APPLICATION ENCRYPTION ARCHITECTURE

White Paper

Contents

Introduction

Organizations and individuals are more connected digitally than ever before. As a consequence, data is accessible to more people than ever before. While digitization accelerates information sharing, it exacerbates the threat of sensitive data falling into the wrong hands. To combat this threat, enterprises turn to encryption.

Encryption is the process of encoding sensitive data so that only authorized parties can read it. Encryption is typically employed in one of these four levels of the technology stack: full-disk (or media-based), file, database, or application. Thales eSecurity provides solutions for encrypting at the file, database, and application layers. The company also offers key management for many full-disk encryption solutions.

When deploying encryption solutions, the lower in the stack the encryption/decryption occurs, the less likely these processes are to interfere with the operations of other layers. For example, if the encryption occurs at the disk level, there is very little risk of the encryption affecting the other layers. The file, database, and application layers will be accessing data in the clear and will function the same as before. On the other hand, if the encryption occurs in the application layer, the disk, file, and database layers will be accessing encrypted data, which may have an impact on operations in these layers. However, because data is encrypted across multiple layers, this approach can reduce the number of potential attack vectors and so increase security.

In spite of these tradeoffs, an organization that wants to establish the highest level of security would be well served by implementing encryption at the application level, which provides the highest level of security with the most granular control of data. However, encryption and key management can be a challenge for many application development organizations. With Vormetric Application Encryption from Thales eSecurity, developers can easily implement application encryption, data access controls, and key management—without having to become cryptography experts or compromise the security of cryptographic keys.

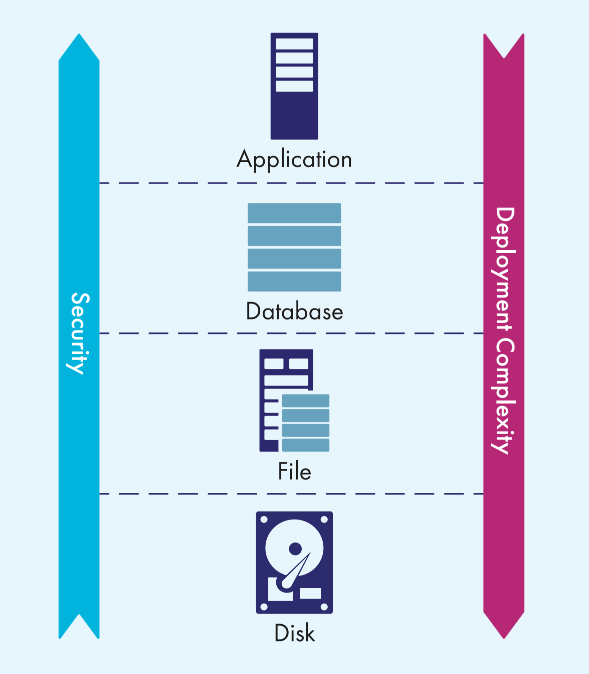

Fig. 1: Security increases when encryption is implemented higher in the stack, but it is more complex to deploy

This paper provides a technical overview of Vormetric Application Encryption. The following sections describe the product's architecture and security model and reveal how it helps organizations comply with security policies and industry standards.

Architecture

Vormetric Application Encryption from Thales eSecurity major components include:

- **Vormetric Data Security Manager (DSM).** The DSM is the central component of the Vormetric Data Security Platform. The DSM offers capabilities for managing policies and keys in a centralized fashion. The DSM stores and manages encryption keys, data access policies, administrative domains, and administrator profiles.
- **Vormetric Application Encryption Library.** Vormetric Application Encryption features a library that implements a subset of PKCS#11 APIs. At runtime the library is a dynamically loaded library (.dll) on Windows or a shared object (.so) on Linux and Unix. The Vormetric Application Encryption library communicates over a secure channel to the DSM.
- **Vormetric Application Encryption RESTful API.** Vormetric Application Encryption is also available as a RESTful application programming interface (API).
- **The Vormetric Tokenization Server.** The Vormetric Tokenization Server receives RESTful API calls and dispatches them to an internal PKCS#11 Vormetric Application Encryption service for processing and response. The Tokenization Server also provides flexible authentication and granular authorization tools.

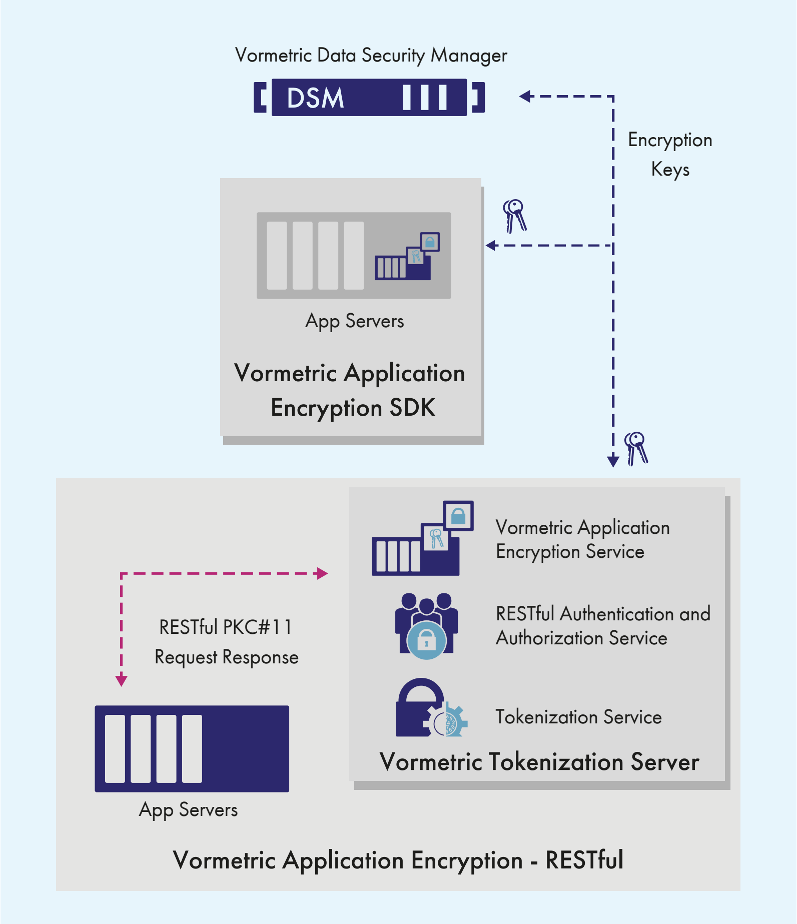
Vormetric Data Security Manager
DSM
Encryption Keys
App Servers
Vormetric Application Encryption SDK
Vormetric Application Encryption Service
RESTful PKC#11 Request Response
RESTful Authentication and Authorization Service
Tokenization Service
Vormetric Tokenization Server
App Servers
Vormetric Application Encryption - RESTful

Fig. 2: The DSM centrally manages policies and keys for the Vormetric Application Encryption Libraries running on servers

Programming languages supported

The Vormetric Application Encryption library supports the following languages:

- **C.** The library is written in C, so C-based functions are exported as part of the library and can be called directly by C- based applications.
- **Java.** The C library can be called from a Java library with the help of a standard JDK package named "sun. security. pkcs11.wrapper". This is a lightweight wrapper provided by the Oracle JDK to allow Java to interface with PKCS#11 C APIs.
- **.NET**. Vormetric Application Encryption ships with an additional DLL for .NET applications. Named "PKCS11InterOp. dll", this DLL converts data types and functions from managed objects in C# to unmanaged C objects in the PKCS#11 library.

Sample code for all three languages is provided with the product.

The Vormetric Application Encryption RESTful API is supported on any server operating system and language that supports RESTful web services.

Standards and APIs

Thales eSecurity is an active voting member of the Oasis PKCS#11 (Public Key Cryptography Standards) open standards committee. Vormetric Application Encryption implements PKCS#11 APIs. The PKCS#11 standard defines a platform- independent API to integrate with cryptographic tokens, smart cards, and hardware security modules (HSMs). The APIs define the most commonly used cryptographic types and operations. The Vormetric application library implements a subset of PKCS#11 APIs to enable the following functions:

- Create a key
- Search for a key by name
- Destroy a key
- Export a key (wrapped with another key) from DSM
- Import a key into DSM
- Encrypt
- Decrypt
- Sign
- Verify

Security model

Vormetric Application Encryption enables development and deployment of customized data security applications. The security mechanisms surrounding Vormetric Application Encryption comprise its Security Model.

Like all components of the Vormetric Data Security Platform based on the Vormetric Data Security Manager (DSM), the base component of the Vormetric Application Encryption security model is separation of duties combined with strong authentication and authorization in the use of encryption keys.

SEPARATION OF DUTIES

Separation of duties is a proven security method for preventing or mitigating the risk of a data breach, by restricting the access held by any one administrator and help prevent any single individual from committing data theft, sabotaging keys, or performing other malicious activities. For this discussion, consider one of the most common use cases for Vormetric Application Encryption: securing data prior to storing it in a database. In this use case, controls are typically divided into at least three, and up to five areas of responsibility:

- Vormetric Data Security Manager Administrators
- Vormetric Tokenization Server Administrators
- Database Administrators
- Application Developers
- System Administrators

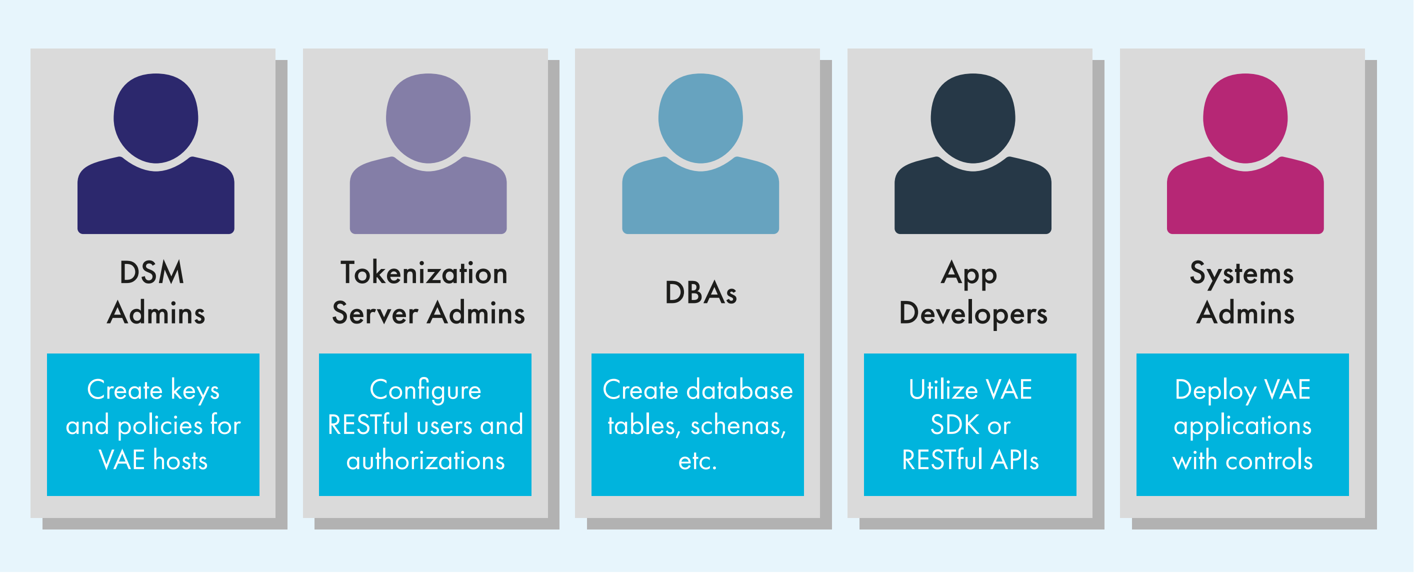
DSM Admins
Create keys and policies for VAE hosts
Tokenization Server Admins
Configure RESTful users and authorizations
DBAs
Create database tables, schenas, etc.
App Developers
Utilize VAE SDK or RESTful APIs
Systems Admins
Deploy VAE applications with controls

Fig. 3: Privileges and responsibilities of each group of users. When implemented properly, no single user can gain complete access to sensitive data without the cooperation of other groups.

Vormetric DSM Administrators

DSM administrators are responsible for the day-to-day administration of the DSM platform. To support DSM multi-tenancy, there is a hierarchy of DSM administrative roles, specifically, DSM System administrators, DSM Domain administrators, and DSM Security administrators. For the purpose of this discussion we focus on DSM Security Administrators, who configure Vormetric Application Encryption hosts and Vormetric Tokenization servers that are allowed to register to a particular DSM or DSM cluster, and create and manage encryption keys and policies. In a good separation of duties environment, no DSM administrators have administrative access to Windows or Linux hosts, nor do they have credentials to administer databases. Consequently, even though DSM Security administrators have access to the encryption keys, they will not be able to gain access to the encrypted data.

Database Administrators (DBAs)

DBAs are responsible for installing, configuring, maintaining, and monitoring databases in an organization. When organizations employ application encryption and separation of duties, they can enable DBAs to control the operational aspects of the database, but without having the privileges needed to access the sensitive data stored within the database.

Application Developers

Application developers work at the top of the encryption stack. They develop applications that use Vormetric Application Encryption APIs to encrypt and decrypt data in databases. To limit their power and privileges, security guidelines often mandate that application developers write and test code using only simulated data. For example, requirement 6.4 of the Payment Card Industry Data Security Standard (PCI DSS) mandates that "The development/test environments are separate from the production environment, with access control in place to enforce the separation.[1]" By enforcing this separation of development and production environments, organizations can ensure that application developers won't be able to decrypt and steal sensitive data.

The simplest way to ensure there is no sensitive data in the simulated data is for the application developer to assist the DBA in using **Vormetric Batch Data Transformation** to copy live database records to a development database while converting all sensitive columns of a database into encrypted or tokenized data.

System Administrators

The role of both the system administrator and application developers might meet at a role such as a security architect, where implementation choices for Vormetric Application Encryption can provide varying levels of security in operation.

Vormetric Tokenization Server Administrators

As the Vormetric Tokenization Server acts as the authentication, authorization, key management and encryption services engine for the RESTful version of Vormetric Application Encryption, its administrator plays a key role in both separation of duties and implementing the security model of Vormetric Application Encryption.

[1] To learn more about the comprehensive PCI DSS coverage that Vormetric provides, please visit: www.thalesesecurity.com/solutions/compliance/global/pci-dss

VORMETRIC APPLICATION ENCRYPTION SECURITY

SDK Security

The Vormetric Application Encryption SDK provides from two to four levels of security. The first two are required. First, any host, either development or production, that utilizes Vormetric Application Encryption must be registered with the Data Security Manager, which provides key management and certain highly secure cryptographic services. Second, during host registration, host certificates are exchanged and a Vormetric Application Encryption "PIN" is input by the system administrator. Servers with the SDK but not registered with the DSM cannot access any key management or cryptographic services.

Host-level security is provided with the PIN. In order to execute the encryption and key management APIs of Vormetric Application Encryption, the user of the application must provide the PIN. Access to the PIN is controlled by the host administrator, not the application developer. The PIN is used to encrypt the private key certificate file used in creating a secure communication channel with the DSM. Every time a Vormetric Application Encryption application starts up, the PIN must be provided to unlock the private key file needed to establish two-way TLS communications with the DSM. If the PIN is lost, the host administrator must re-register the Vormetric Application Encryption host with the DSM and create a new PIN. When developing applications, the PIN is used in the login SDK command.

A third, optional security level offers identity-based access to encryption keys or groups of keys. On the Vormetric Data Security Manager, keys are assigned to key groups. A key group can hold as few as one key. "Identities" consist of usernames and passwords. One or more user names are assigned to key groups. Then, when an application provides a user name and password using the login SDK command, the application and username gain access to, but is also limited to, keys in the assigned groups. The benefits of identity-based access to keys in Vormetric Application Encryption include:

- Multiple applications on a single server get granular, identity-based access to keys
- Application on any server can gain access only to keys in their assigned group

A fourth, optional security level, but again at the host, rather than application level,utilizes hardware-specific information when creating the host certificate used to establish communications with the DSM. Hardware association adds to the host certificate various hardware attributes of the host it is installed on, including the MAC address. With hardware association, even if a rogue developer steals both certificates and the PIN from the installed host, they will not be able to simply copy the certificates and connect to the DSM with a rogue machine.

RESTful API Security

As noted above, the Vormetric Tokenization Server fields request for Vormetric Application Encryption RESTful API calls, forwarding them to the local Vormetric Application Encryption service. The Tokenization Server offers multiple levels of application security. First, RESTful API calls require either a user name and password or client certificate, which must be known to the Tokenization Server either in a local database or in a network server such as Active Directory or LDAP. Second, for each user or group they are a member of, the tokenization server provides granular access control on a per-key basis. Each key has controls for key operations:

Encryption Operations

- Encryption
- Decryption
- Sign
- Verify

Key Management Operations

- Destroy
- Modify
- Export
- Find

The controls are assigned on a per-user or per-group basis.

ENCRYPTION

The Vormetric Application Encryption library offers capabilities for AES Encryption, supporting 128 and 256 bit keys in CBC mode, and format-preserving encryption (FPE).

AES is an encryption standard established by the National Institute of Standards and Technology (NIST) in 2001. Since then, AES has become the industry standard for encryption. The 256-bit key size is considered, cryptographically, to be sufficiently secure.

FPE was approved by NIST in early 2016 and it leverages AES algorithms and functionality. Unlike with most other encryption algorithms, with FPE the resulting ciphertext isn't any longer than the original value of the plaintext that was encrypted. This feature can be useful when the contents of fixed-size database columns, like those containing credit card numbers or Social Security numbers, need to be encrypted, without modifying the schema of the database.

CENTRALIZED KEY MANAGEMENT

Enterprises must not only protect against data theft, but they must also protect their encryption keys from theft, misplacement, and accidental destruction. To facilitate these safeguards, Vormetric Application Encryption supports enterprise key management through the DSM. The DSM is available as a secured, FIPS 140-2 Level 2 or 3-certified appliance for on-premises deployment, or a FIPS 140-2 Level 1-certified virtual appliance. Like other Vormetric Data Security Platform products, all keys that are created and used by the Vormetric Application Encryption library reside in the DSM. This gives customers a unified, centralized platform for encryption and key management.

APPLICATION-LEVEL ENCRYPTION IMPLICATIONS

As mentioned above, encrypting data at the application level presents implementation choices that need to be considered in advance. Most pertain to encrypting data for highly structured storage such as databases. There are three main considerations:

1. **Data Size Changes Required by AES-CBC Encryption** Use of AES-based encryption mechanisms require two changes to the database schema for each field to be encrypted.
   a. The data type changes from the original format (e.g. integer, varchar) to binary
   b. The data size expands, comprised of the original data size plus the block size (16 bytes)

   If it is not possible to change the database schema, consider using one of the several Format-Preserving Encryption (FPE) mechanisms. FPE requires multiple AES operations internally, so encryption performance of FPE may be lower than with AES. The balance between convenience and performance may be irrelevant if, for example, new records to be encrypted arrive at a relatively well-understood rate.
2. **Tracking the AES Initialization vector or the FPE Tweak** Best practices for structured data (i.e. database) encryption state that a unique initialization vector (IV) for AES encryption or "tweak" for FPE should be created and stored for each data row. This ensures that when identical data appears in multiple rows of a column, encrypted values will be different for each, eliminating patterns through which algorithmic attacks might gain information. Storing the IV or tweak requires a database schema change.

3. **Key Rotation and Automated Key Versioning**
Many security mandates, including PCI DSS, require that encryption keys be rotated every 12 to 24 months. Other best practices suggest rotating keys more frequently to reduce the "blast radius" of any single compromised key. Key rotation can complicate key management in applications. To drastically reduce application key management complexity, Vormetric Application Encryption offer automated key versioning. Automated key versioning creates new key material for a named key at regular intervals specified by the application writer or by the administrator in charge of the Vormetric Data Security Manager. Automated key versioning reduces the number of unique key names required to fulfill key rotation mandates by enabling indefinite reuse of a single key name.

The software and management convenience of Automated Key Versioning, however, requires a header prepending all returned ciphertext, and under almost all circumstances, the "key header" must be stored with the encrypted data.

It is possible, as well, to implement key rotation without Automated Key Versioning. The developer writes code to create new keys at a defined rotational interval and stores the key name with all data so that the proper key may be retrieved for decryption.

Performance considerations

This section pertains to the SDK version of Vormetric Application Encryption. To maintain the highest level of security, Vormetric Application Encryption can be configured to ensure encryption keys never leave the DSM. In this scenario, the data that must be encrypted or decrypted is sent to the DSM. However, this mode should be used only when small data objects, such as certificates or passwords, must be encrypted or decrypted. For organizations that have to support the encryption of large data sets and data in performance-sensitive, highly transactional operations, Vormetric Application Encryption provides an optional local key cache. Keys are securely exported from the DSM and then stored locally within the Vormetric Application Encryption library. Subsequent cryptographic operations are performed locally without going to the DSM, which offers significant performance advantages. Using this approach, organizations can perform 50,000 transactions per second, per thread. Certain encryption modes require the key to be cached on the host. All rules pertaining to encryption and key caching are described in detail in the API and the product manual.

Example application encryption workflow

Consider this common workflow in an e-commerce scenario:

1. The user submits personal information to purchase items. Data is secure from the user to the web server.
2. The web server typically decrypts data and sends personal information to the application server in clear text.
3. Vormetric Application Encryption offers two usage models as depicted in figures 4a and 4b.

4a. The application calls into the Vormetric Application encryption local library to encrypt selected sensitive data. The local library returns encrypted values back to the application.

4b. T he application uses a RESTful API to contact the Vormetric Application Encryption Service in the Vormetric Tokenization Server. The service returns encrypted values back to the application.

5. The application forwards or stores encrypted data.

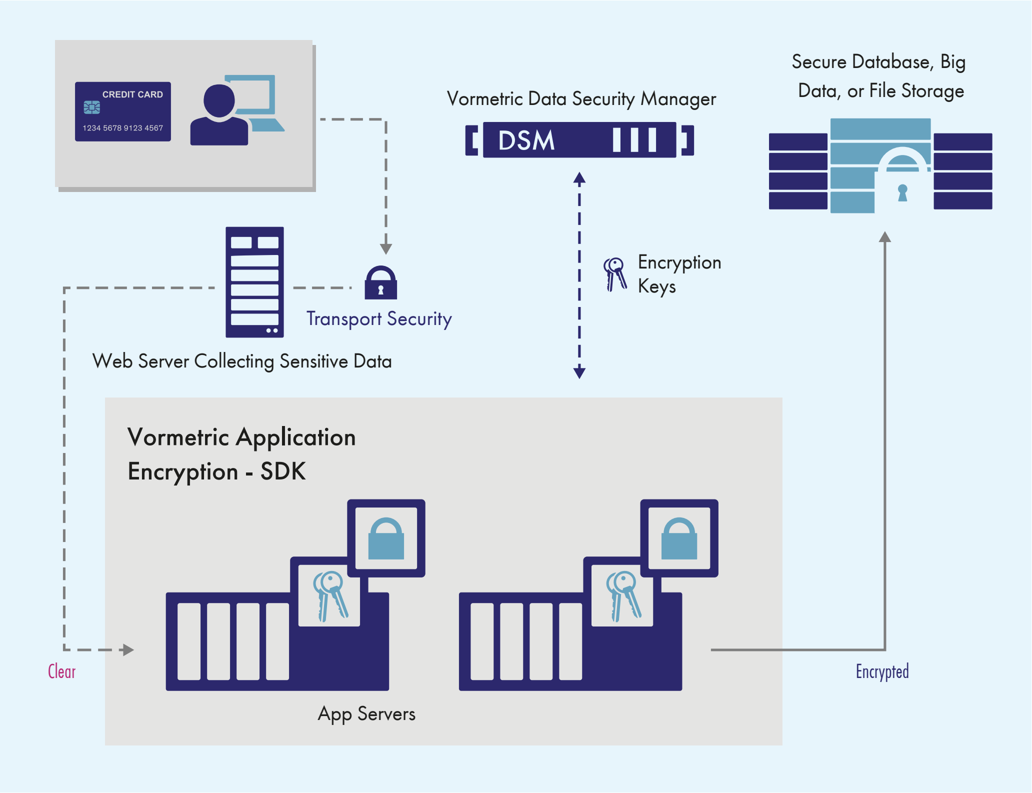
CREDIT CARD
1234 5678 9123 4567
Vormetric Data Security Manager
DSM
Secure Database, Big Data, or File Storage
Encryption Keys
Transport Security
Web Server Collecting Sensitive Data
Vormetric Application Encryption - SDK
Clear
App Servers
Encrypted

Fig. 4a: Example application workflow, using SDK

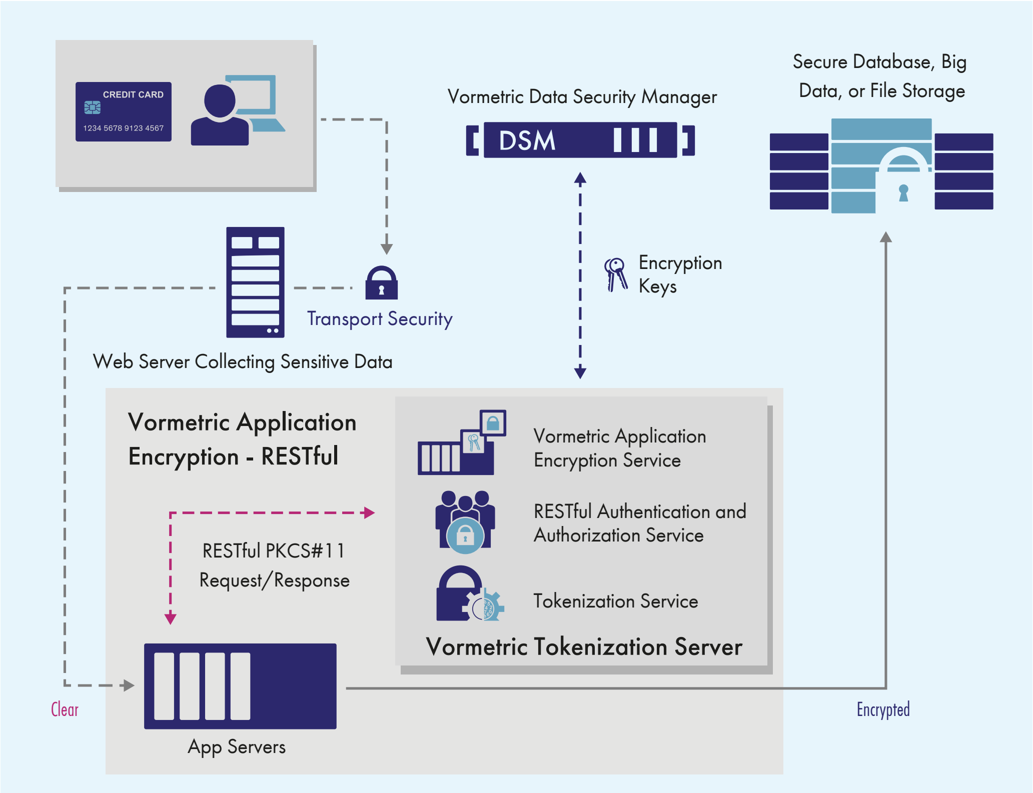
CREDIT CARD
1234 5678 9123 4567
Vormetric Data Security Manager
DSM
Secure Database, Big Data, or File Storage
Encryption Keys
Transport Security
Web Server Collecting Sensitive Data
Vormetric Application Encryption - RESTful
Vormetric Application Encryption Service
RESTful Authentication and Authorization Service
RESTful PKCS#11 Request/Response
Tokenization Service
Vormetric Tokenization Server
Clear
App Servers
Encrypted

Fig. 4b: Example application workflow, using RESTful API's

Operational aspects

APPLICATION INTEGRATION

Thales eSecurity customers can download the Vormetric Application Encryption sample code collection. The sample code collection has a wide range of sample code for both typical PKCS#11 operations and for using Vormetric Application Encryption advanced features such as key versioning, Sample code is available in C, C-Sharp and Java with sample applications that utilize the RESTful API.

UPGRADES AND PATCHING

Newer versions of the DSM support older versions of the Vormetric Application Encryption agent, but not vice versa. Consequently, the recommended workflow is for customers to upgrade their DSMs first, and then upgrade their existing Vormetric Application Encryption agents.

Vormetric Application Encryption libraries can be easily upgraded to newer versions without having to re-register with the DSM, which helps ensure business continuity.

Conclusion

Application-layer encryption is arguably the most secure form of encryption because it runs at the top of the technology stack. Vormetric Application Encryption is based on the PKCS#11 standard, which has been widely used and proven for many years. With the solution's APIs, enterprises can easily integrate application encryption into their existing applications. Together with the Vormetric Data Security Manager, Vormetric Application Encryption reduces significantly the complexity and risk of implementing an in-house encryption and key management solution.

THALES

www.thalesesecurity.com

About Thales eSecurity

Thales eSecurity is a leader in advanced data security solutions and services that deliver trust wherever information is created, shared or stored. We ensure that the data belonging to companies and government entities is both secure and trusted in any environment – on-premises, in the cloud, in data centers or big data environments – without sacrificing business agility. Security doesn't just reduce risk, it's an enabler of the digital initiatives that now permeate our daily lives – digital money, e-identities, healthcare, connected cars and, with the internet of things (IoT), even household devices. Thales provides everything an organisation needs to protect and manage its data, identities and intellectual property, and meet regulatory compliance – through encryption, advanced key management, tokenisation, and privileged-user control and high-assurance solutions. Security professionals around the globe rely on Thales to confidently accelerate their organization's digital transformation. Thales eSecurity is part of Thales Group.

Follow us on:

Americas – Thales eSecurity Inc. 900 South Pine Island Road, Suite 710, Plantation, FL 33324 USA • Tel:+1 888 744 4976 or +1 954 888 6200 • Fax:+1 954 888 6211 • E-mail: sales@thalesesec.com
Asia Pacific – Thales Transport & Security (HK) Lt, Unit 4101-3, 41/F, Sunlight Tower, 248 Queen's Road East, Wanchai, Hong Kong • Tel:+852 2815 8633 • Fax:+852 2815 8141 • E-mail: asia.sales@thales-esecurity.com
Europe, Middle East, Africa – Meadow View House, Long Crendon, Aylesbury, Buckinghamshire HP18 9EQ • Tel:+44 (0)1844 201800 • Fax:+44 (0)1844 208550 • E-mail: emea.sales@thales-esecurity.com