SDC 18
September 24-27, 2018
Santa Clara, CA
www.storagedeveloper.org

A KMIP User Guide

How to Secure Applications for Today's Cybersecurity Environment

John Leiseboer
QuintessenceLabs

A KMIP User Guide

1. KMIP: What it is and how it Works
2. Use Cases and Vendors
3. Capabilities and Benefits
4. Things to Know Before Deployment
5. Deep Dive
6. Steps to Get Started

KMIP: Overview

- **What Is It?**
  - KMIP = **K**ey **M**anagement **I**nteroperability **P**rotocol
  - Standard for managing the creation, distribution and lifecycle of cryptographic objects
  - Part of OASIS (**O**rganization for the **A**dvancement of **S**tructured **I**nformation **S**tandards)

- **What Does it Do?**
  - Single, comprehensive protocol for communication between encryption systems
    - Email; databases; storage devices…
    - Supports encryption keys, certificates, secret data, split keys, wrapped keys, and more
  - Supports deployment of integrated key management and encryption across an organization

Will provide better data security, enable more pervasive encryption and reduce costs by removing redundant, incompatible key management processes

www.oasis-open.org/committees/tc_home.php?wg_abbrev=kmip

OASIS KMIP Timelines

- OASIS KMIP is a standard for key management interoperability

2007 Standard Key Management Protocol (SKMP) specification by EMC/RSA, IBM, nCipher and HP

2009 SKMP renamed KMIP and moved to OASIS

2010 KMIP v1.0 OASIS specification published

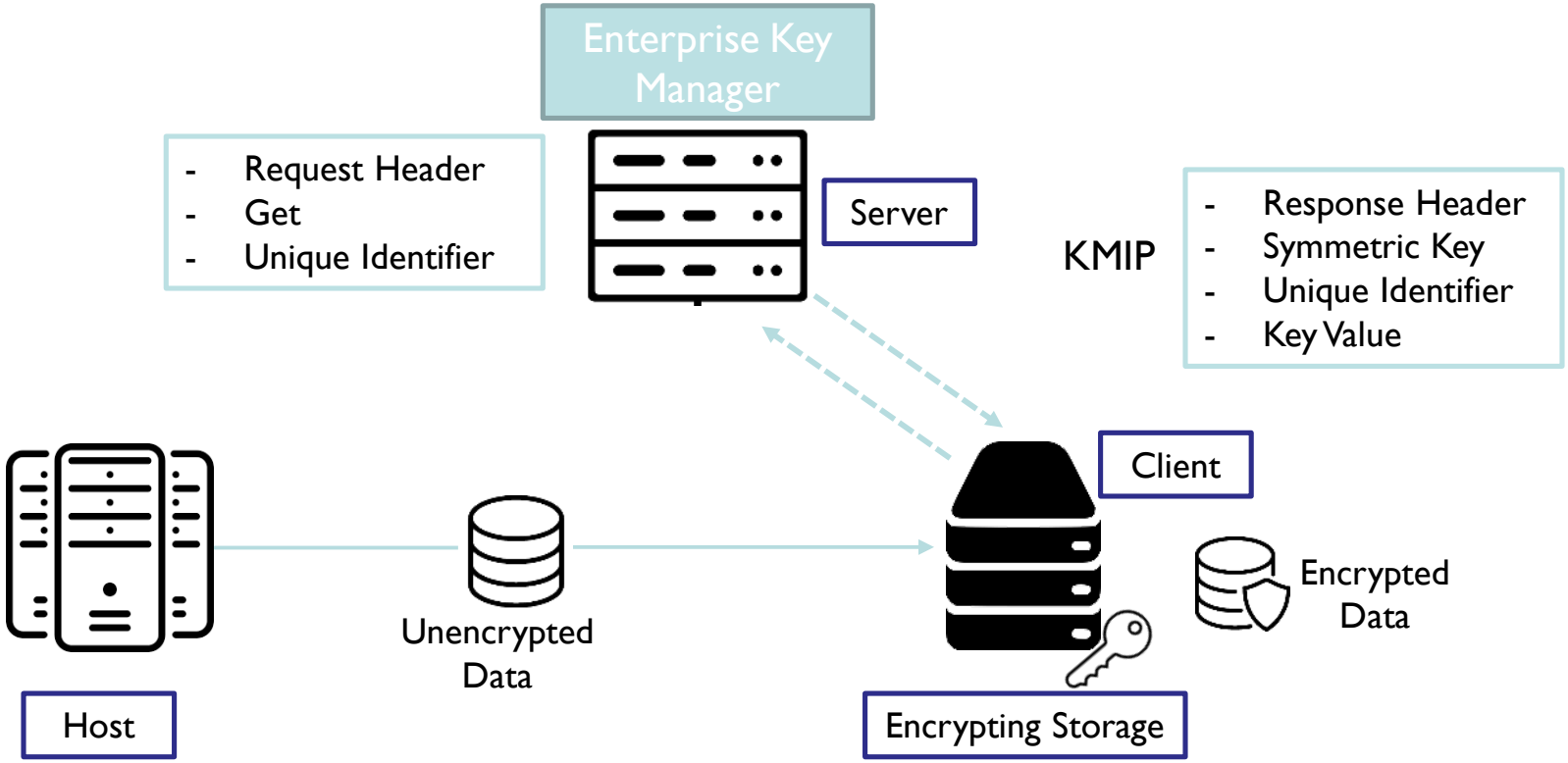
2013 KMIP v1.1 OASIS specification published

2016 KMIP v1.3 OASIS specification published

2017 KMIP v1.4 OASIS specification published

- Widely adopted, suited for many use cases, including embedded and IoT

KMIP: How it Works



KMIP: Use Cases and Vendor Interop

Use Cases

- Managed Service
- Virtual Zeroization
- Application Encryption
- Link Encryption
- Database Encryption
- File Encryption
- SAN Encryption
- Disk Array Encryption
- Tape Encryption

Information Protection

VM
SSH Public Keys
X509
Certificates
Smart Card
PostgreSQL
MS SQL Server
HD Video
VSAN
Disk Array
Tape

- Key Management (qCrypt)
- Entropy Source (qStream)
- Hardware Security Module

Trusted Security Foundation

Vendors

ORACLE DATABASE, Microsoft, SafeNet, hp, Quintessence Labs, PKWARE, vmware, IBM, netdocuments, Vormetric, THALES, NetApp

OASIS

Client/End User Application
KMIP
PKCS#11

KMIP: Capabilities

Protocol Operations >30

- Create
- Create Key Pair
- Register
- Re-Key
- Re-key Key Pair
- Derive Key
- Certify
- Recertify
- Locate
- Check
- Get
- Get Attributes
- Get Attributes List
- Add Attribute
- Modify Attribute
- Delete Attribute
- Obtain Lease
- Get Usage Allocation
- Activate
- Revoke
- Destroy
- Archive
- Recover
- Validate
- Query
- Version
- Discover
- Cancel
- Poll
- Notify
- Put
- and more…

Managed Objects >10

- Certificate
- Symmetric Key
- Public Key
- Private Key
- Split Key
- Template
- Policy Template
- Secret Data
- Opaque Object
- Key Block (for keys) or Value (for certificates)

Object Attributes >40

- Unique Identifier
- Name
- Object Type
- Cryptographic Algorithm
- Cryptographic Length
- Cryptographic Parameters
- Cryptographic Domain Parameters
- Certificate Type
- Certificate Length
- X.509 Certificate Identifier
- X.509 Certificate Subject
- X.509 Certificate Issuer
- Certificate Identifier
- Certificate Subject
- Certificate Issuer
- Link
- Digital Signature Algorithm
- Contact Information
- Last Change Date
- Custom Attribute
- Digest
- Operation Policy Name
- Cryptographic Usage Mask
- Lease Time
- Usage Limits
- State
- Initial Date
- Activation Date
- Process Start Date
- Protect Stop Date
- Deactivation Date
- Destroy Date
- Compromise Occurrence Date
- Compromise Date
- Revocation Reason
- Archive Date
- Object Group
- Fresh
- Application Specific Information
- and more…

Benefits of KMIP

- Simplifies key management integration
- Facilitates adoption of strong enterprise-level encryption
- Helps avoid siloed encryption and associated vulnerabilities
- Common set of instructions for working with cryptographic objects
- Powerful: supports large number of Operations, Objects and Attributes
- Verified annually with large number of vendors

KMIP is the best standard for strong data protection implementations across multiple devices

Implementing KMIP – Tips and Tricks

- As any standard, KMIP has some limitations
  - These do not prevent implementation of Key Management solutions across multiple vendors
  - They require some deeper knowledge to implement

- Goals of next section:
  - Provide you some tips and insights to successfully deploy KMIP
  - NOT scare you off your KMIP deployment!

KMIP is the best standard for strong data protection implementations across multiple devices

KMIP Tips and Tricks: Useability Challenges

- Locate can return nothing
  - The result of a search is allowed to be “nothing found”
  - “Nothing found” is indicated by an empty response
  - Need to be aware of this, especially when performing batch operations
- Templates
  - A convenient means of grouping together attributes that can be applied consistently to managed objects
  - Has found use as part of object policies in some KMS products
  - Templates used extensively in deployed systems
  - The Tech Committee deprecated templates in 1.2 citing inconsistent vendor implementations in KMIP 1.0
- Redundant fields in Password Based Key Derivation Function
  - PBKDF specification says that derivation data MUST be included in the request UNLESS a UID (pointing to a Secret Data object) is provided; => UID is Optional
  - Be careful: the Derive Operation always requires a UID

Usability Example: Locate can return nothing

```
$ s_find -match "Name=JOHNL-SKEY-001"
QuintessenceLabs Client SDK Sample

Connection established with server
---------------
Find Object...
Response received
UID: 9d345611-e9bc-4010-8c52-4dc188501c25
Find Object finished
```

Locate object that **exists**

UID of object is returned

```
$ s_find -match "Name=JOHNL-XKEY-001"
QuintessenceLabs Client SDK Sample

Connection established with server
---------------
Find Object...
Response received
Find Object finished
```

NOTHING

Locate object that **doesn't exist**

Empty result is returned

Beware of empty response, particularly for batch operations

KMIP Tips and Tricks: Size Limits and Offsets

- Locate can be implemented with Limits and Offset.

  "The request MAY contain a Maximum Items field… The request MAY contain an Offset Items field…
  "If both Offset Items and Maximum Items are specified in the request, the server skips Offset Items objects and returns up to Maximum Items objects."

  This works well for a single client and single connection…

- But…
  - KMIP is a stateless protocol and supports multiple clients and multiple concurrent client connections
  - Locate with Limits and Offset can break in real world with multiple clients and/or connections running

Iterate through list of UIDs and destroy the keys…

```
$ s_skey -destroy -uid '096a2bf6-2fec-4ad1-8f48-188b3c6cc233'
QuintessenceLabs Client SDK Sample
Connection established with server
---------------
Destroy Symmetric Key...
Request failed, error: 1, reason: Server Error:Permission Denied, Explanation:Object already destroyed
destroy_skey() failed, error: 1
```

Example of error message linked to limits and offset challenges

Interoperability: Message Size Limit Solutions

- Single client, single thread
- Don’t use Locate, or don’t rely on Limit and Offset
- Lots of error handling code to try to handle the problem
- Need to cope with changed order of values, omitted values, duplicated values, increasing and decreasing number of values

Or better:

- KMIP improvement (fixed in 2.0?)
- Use of a KMIP extension
  - Extensions are allowed but are usually vendor specific
  - User needs to identify client and server vendors that support an extension to fix this flaw

KMIP Challenges: Security

- Random Number Generation
  - Any client is permitted to seed the server RNG

  → Potential for a client to control another client's random numbers and key values

- Cryptographic Operations
  - Client can force server to perform cryptographic operations that violate permitted operating parameters for managed keys

  → Example: a key may require crypto to be performed in CBC mode only, but the client can instruct the server to perform crypto in ECB mode

  Note: This is explicitly disallowed for key wrapping performed on the server

Solutions: Implement appropriate controls; needs in-depth knowledge of protocol capabilities

Security: Random

RNG Algorithm	
Name	**Value**
Unspecified	00000001
FIPS 186-2	00000002
DRBG	00000003
NRBG	00000004
ANSI X9.31	00000005
ANSI X9.62	00000006
Extensions	8XXXXXXX

RNG Mode	
Name	**Value**
Unspecified	00000001
Shared Instantiation	00000002
Non-Shared Instantiation	00000003
Extensions	8XXXXXXX

KMIP Tips and Tricks: Standards Compliance

- NIST SP 800-57 Part 1 is a normative reference to KMIP

  But… some KMIP test cases specify responses that violate NIST SP 800-57 Part 1 requirements (Asymmetric key lifecycle operations, Derived key lifecycle operations)

- TTLV is a required message encoding
  - Test cases are specified using XML, an optional binding
  - No DTD or schema specified for the XML bindings

Thoughtful implementation is needed to fully comply with standards!

KMIP Challenges: Performance

- KMIP is mostly a stateless protocol: the server is not required to retain session information for the duration of multiple requests
- This partly changed with KMIP 1.3: Cryptographic streaming operations require the server to maintain state
  - No state management provisions are currently in the standard (now at 1.4)
    - What happens if a stream is interrupted? How is it recovered?
    - What happens if a stream is prematurely terminated?
    - How does a server know if a stream has failed, or is just waiting?
  - The streaming protocol is inefficient
    - Full round-trip time latency between each stream part
    - No possibility to pipeline messages

If high performance is an issue, you may need to perform crypto in the client, or use a different network protocol instead of KMIP.

KMIP: What next?

- ✓ OASIS KMIP
  - ✓ Follow what's going on
  - ✓ Comment on proposed standards
  - ✓ Join and participate
- ✓ Lobby your vendors to fully implement KMIP and participate in process
- ✓ Get informed: At the very least read the standards documents - "caveat emptor"
- ✓ **Ask Questions!**

The phrase *caveat emptor* arises from the fact that buyers often have less information about the good or service they are purchasing, while the seller has more information. Defects in the good or service may be hidden from the buyer, and only known to the seller. Thus, the buyer should beware.* * http://en.wikipedia.org/wiki/Caveat_emptor