SDC
STORAGE DEVELOPER CONFERENCE
SNIA ■ SANTA CLARA, 2015

Multi-Vendor Key Management with KMIP

Tim Hudson
CTO & Technical Director
CRYPTSOFT
tjh@cryptsoft.com

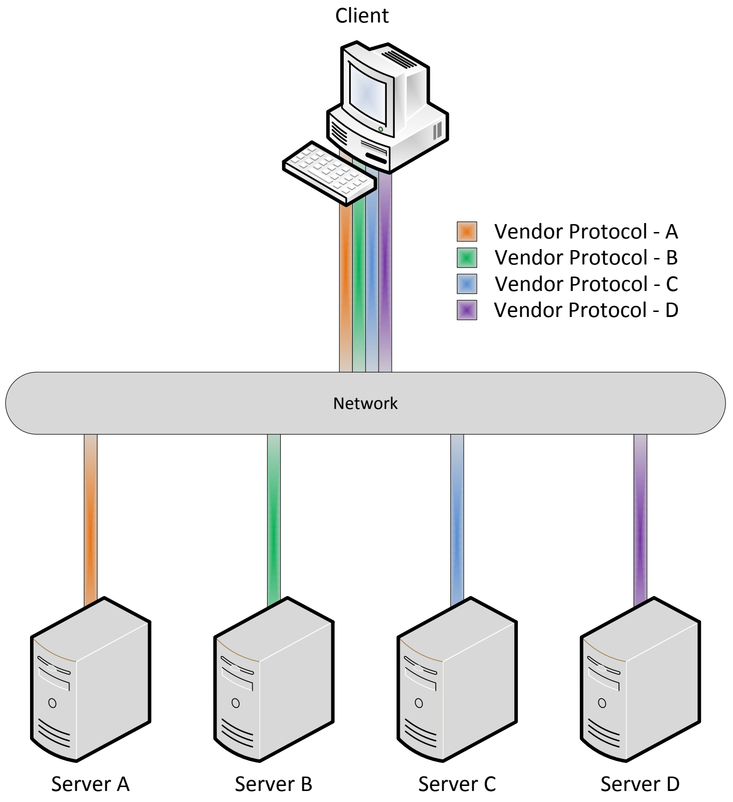
Abstract

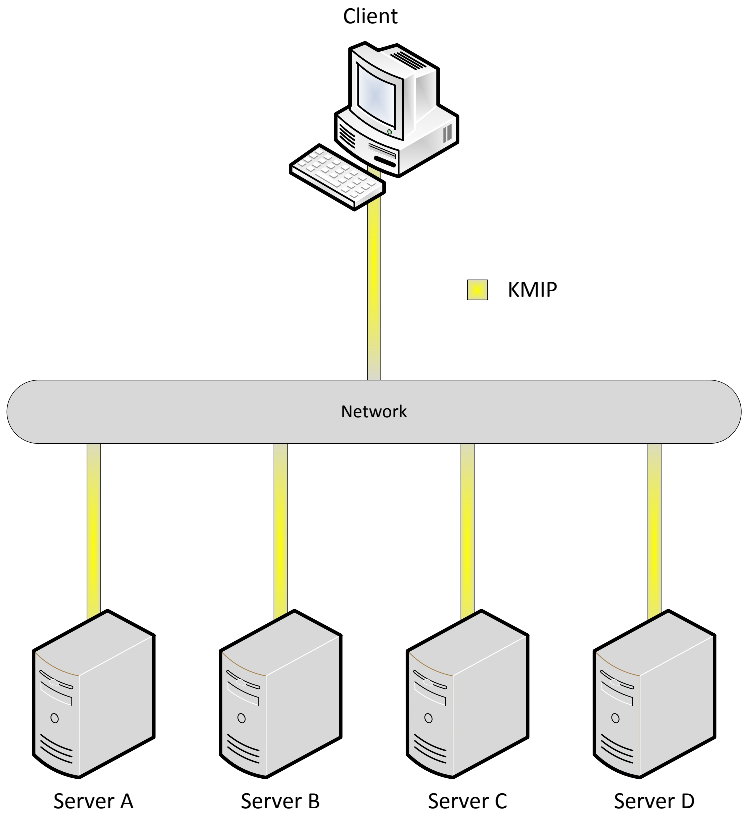
- Practical experience from implementing KMIP and from deploying and interoperability testing multiple vendor implementations of KMIP.
- Guidance covering the key issues you need to ensure that your vendors address
- How to distinguish between simple vendor tick-box approaches to standard conformance and actual interoperable solutions.

The need for multi-vendor

KEY MANAGEMENT

Multi-Vendor – Single Integration





Prior to KMIP each application had to support each vendor protocol

With KMIP each application only requires support for one protocol

Multi-Vendor – Single Integration

- Positive
  - Single Integration with single SDK
  - Common vocabulary
  - Greater choice of technology providers
  - “Free” interoperability without point-to-point testing
- Negative
  - Have to actually follow a standard
  - Vocabulary may not match current usage
  - May need to implement more than is strictly necessary
  - No control over end-user integration

Real-world usage of OASIS KMIP

KEY MANAGEMENT

Multi-Vendor – Who and Where

Storage

- Disk Arrays, Flash Storage Arrays, NAS Appliances
- Tape Libraries, Virtual Tape Libraries
- Encrypting Switches
- Storage Key Managers
- Storage Controllers
- Storage Operating Systems

Infrastructure

- Key Managers
- Hardware security modules
- Encryption Gateways
- Virtualization Managers
- Virtual Storage Controllers
- Network Computing Appliances

Cloud

- Key Managers
- Compliance Platforms
- Information Managers
- Enterprise Gateways and Security
- Enterprise Authentication
- Endpoint Security

BDT BROCADE DATASTAX panzura SEPATON
ETINET Quantum RSA The Security Division of EMC NetApp Hitachi Data Systems
IBM DELL The power to do more hp THALES FORNETIX Vormetric SafeNet
VENAFI ORACLE BRACKET COMPUTING HUAWEI CipherCloud Trust in the Cloud TREND MICRO
HYTRUST Cloud Under Control Townsend SECURITY rsd Your Information. Governed. FUJITSU skyhigh netskope

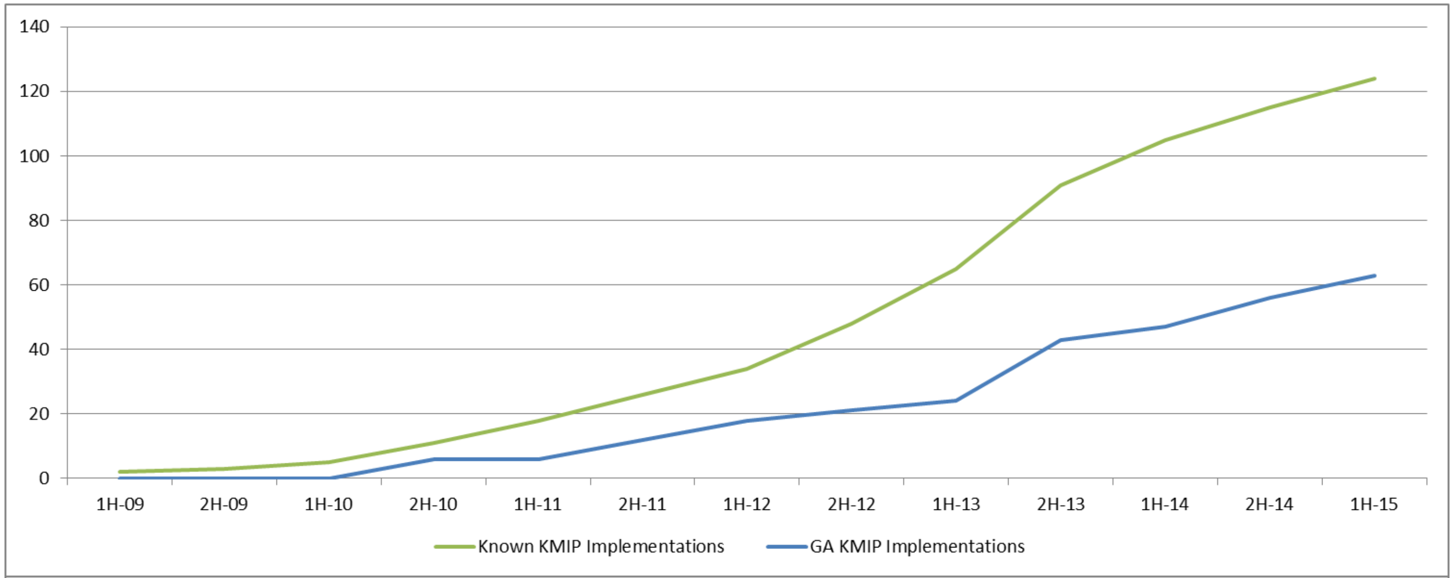
Multi-Vendor – What

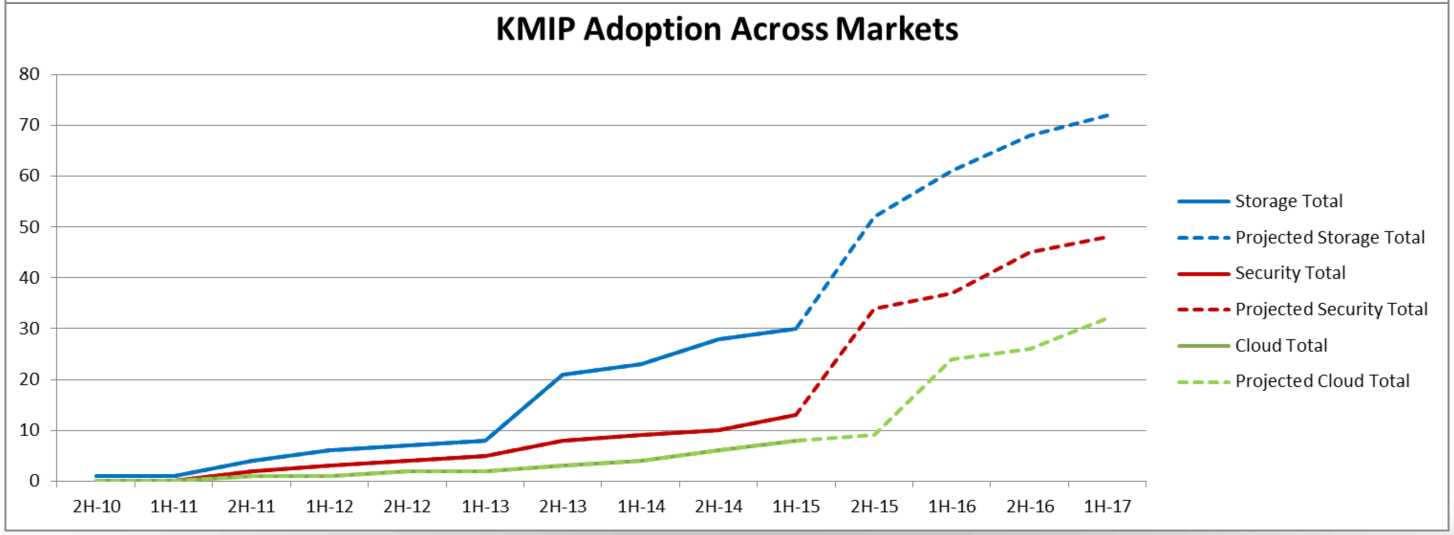
- Disk Arrays, Flash Storage Arrays, NAS Appliances, Storage Operating Systems
  - Vaulting master authentication key
  - Cluster-wide sharing of configuration settings
  - Specific Usage Limits checking (policy)
  - FIPS140-2 external key generation (create, retrieve)
  - Multi-version key support during Rekey
  - Backup and recovery of device specific key sets

Multi-Vendor – What

- Tape Libraries, Virtual Tape Libraries
  - External key generation (create, retrieve)
  - FIPS140-2 external key generation (create, retrieve)
  - Multi-version key support during Rekey
- Encrypting Switches, Storage Controllers
  - Vaulting device or port specific encryption keys
  - Cluster-wide sharing of configuration settings
  - Specific Usage Limits checking (policy)

Multi-Vendor – How many products





OASIS KMIP

SPECIFICATION

OASIS KMIP Specification

- OASIS KMIP 1.0 – Oct 2010

Specification	105 pages	
Profiles	16 pages	
Usage Guide	44 pages	
Use Cases (Test Cases)	168 pages	

- OASIS KMIP 1.1 – Jan 2013

Specification	164 pages	+56%
Profiles	39 pages	+143%
Usage Guide	63 pages	+43%
Test Cases	513 pages	+205%

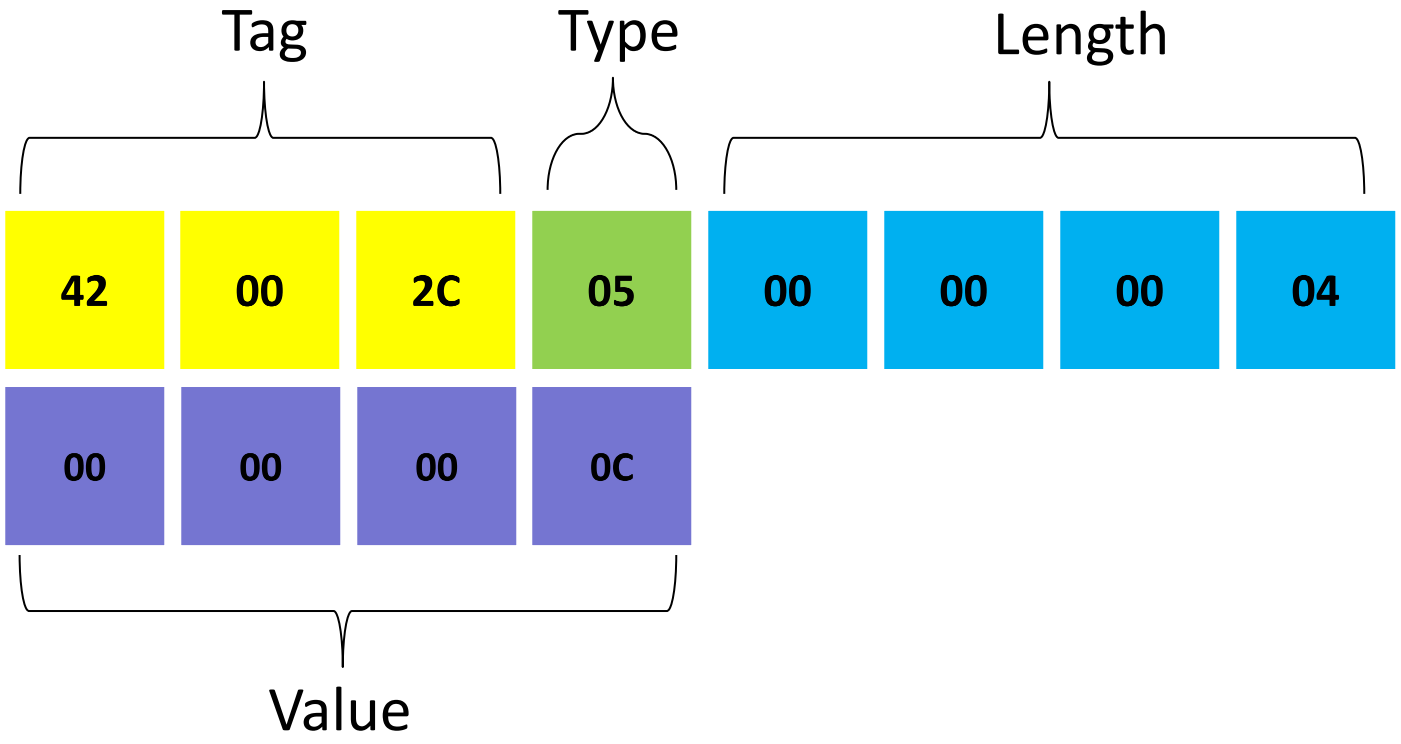
- OASIS KMIP 1.2 –May 2015

Specification	188 pages	+14%
Profiles (multiple)	871 pages	+2133%
Usage Guide	78 pages	+24%
Test Cases	880 pages	+70%
Use Cases	130 pages	

KMIP fundamentals

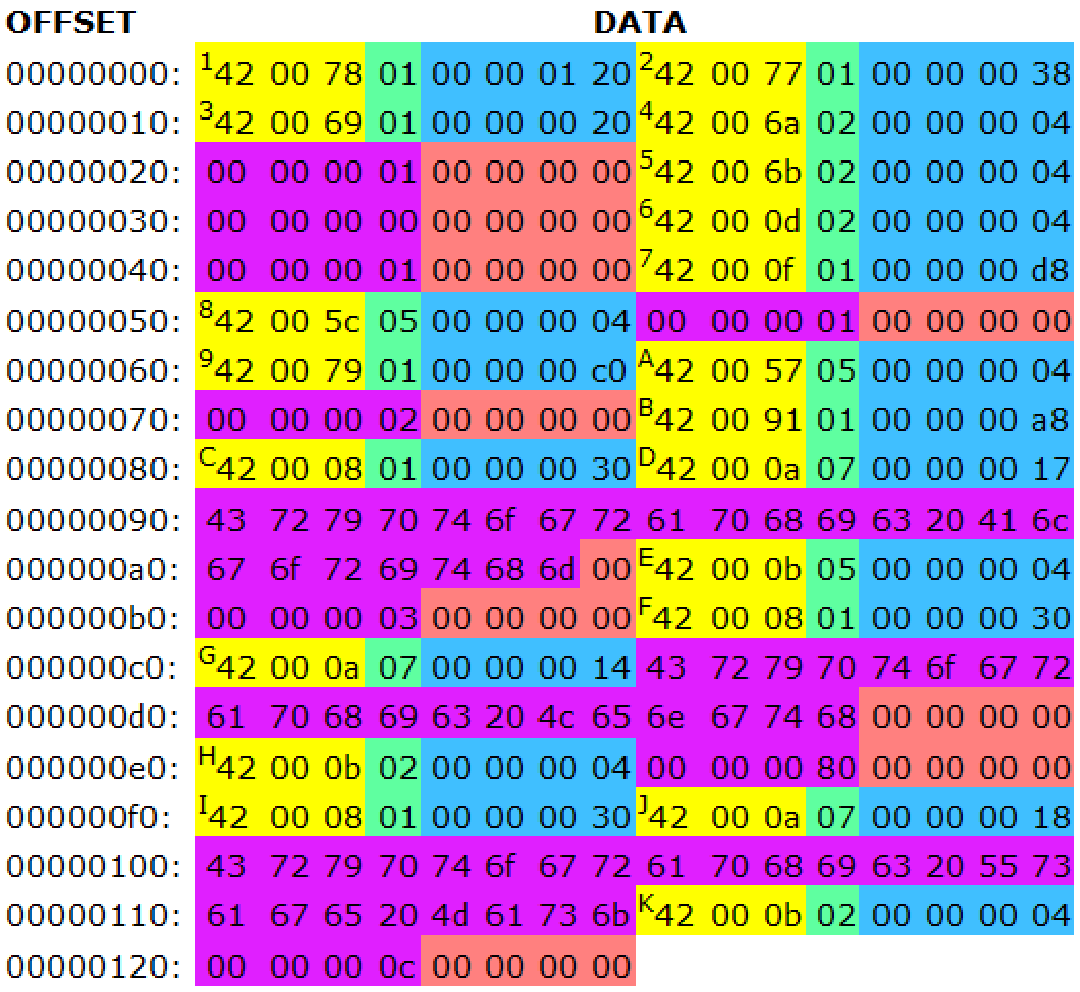
Message Encoding

- Binary Tag-Type-Length-Value format
- Optional JSON and XML encoding in KMIP$^{1.2}$



Cryptographic Usage Mask = Encrypt | Decrypt

KMIP fundamentals



```
OFFSET                      DATA
00000000: [1]42 00 78 01 00 00 01 20 [2]42 00 77 01 00 00 00 38
00000010: [3]42 00 69 01 00 00 00 20 [4]42 00 6a 02 00 00 00 04
00000020:    00 00 00 01 00 00 00 00 [5]42 00 6b 02 00 00 00 04
00000030:    00 00 00 00 00 00 00 00 [6]42 00 0d 02 00 00 00 04
00000040:    00 00 00 01 00 00 00 00 [7]42 00 0f 01 00 00 00 d8
00000050: [8]42 00 5c 05 00 00 00 04    00 00 00 01 00 00 00 00
00000060: [9]42 00 79 01 00 00 00 c0 [A]42 00 57 05 00 00 00 04
00000070:    00 00 00 02 00 00 00 00 [B]42 00 91 01 00 00 00 a8
00000080: [C]42 00 08 01 00 00 00 30 [D]42 00 0a 07 00 00 00 17
00000090:    43 72 79 70 74 6f 67 72    61 70 68 69 63 20 41 6c
000000a0:    67 6f 72 69 74 68 6d 00 [E]42 00 0b 05 00 00 00 04
000000b0:    00 00 00 03 00 00 00 00 [F]42 00 08 01 00 00 00 30
000000c0: [G]42 00 0a 07 00 00 00 14    43 72 79 70 74 6f 67 72
000000d0:    61 70 68 69 63 20 4c 65    6e 67 74 68 00 00 00 00
000000e0: [H]42 00 0b 02 00 00 00 04    00 00 00 80 00 00 00 00
000000f0: [I]42 00 08 01 00 00 00 30 [J]42 00 0a 07 00 00 00 18
00000100:    43 72 79 70 74 6f 67 72    61 70 68 69 63 20 55 73
00000110:    61 67 65 20 4d 61 73 6b [K]42 00 0b 02 00 00 00 04
00000120:    00 00 00 0c 00 00 00 00
```

TTLV Encoding

KMIP fundamentals

```xml
<RequestMessage>
  <RequestHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="0"/>
    </ProtocolVersion>
    <BatchCount type="Integer" value="1"/>
  </RequestHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Create"/>
    <RequestPayload>
      <ObjectType type="Enumeration" value="SymmetricKey"/>
      <TemplateAttribute>
        <Attribute>
          <AttributeName type="TextString" value="Cryptographic Algorithm"/>
          <AttributeValue type="Enumeration" value="AES"/>
        </Attribute>
        <Attribute>
          <AttributeName type="TextString" value="Cryptographic Length"/>
          <AttributeValue type="Integer" value="128"/>
        </Attribute>
        <Attribute>
          <AttributeName type="TextString" value="Cryptographic Usage Mask"/>
          <AttributeValue type="Integer" value="Decrypt Encrypt"/>
        </Attribute>
      </TemplateAttribute>
    </RequestPayload>
  </BatchItem>
</RequestMessage>
```

XML Encoding - KMIP$^{1.2}$

KMIP Fundamentals

- Managed Objects have a “Value”
  - Value is set at object creation
  - Value cannot be changed
  - Value may be “incomplete”
  - Value may be in varying formats

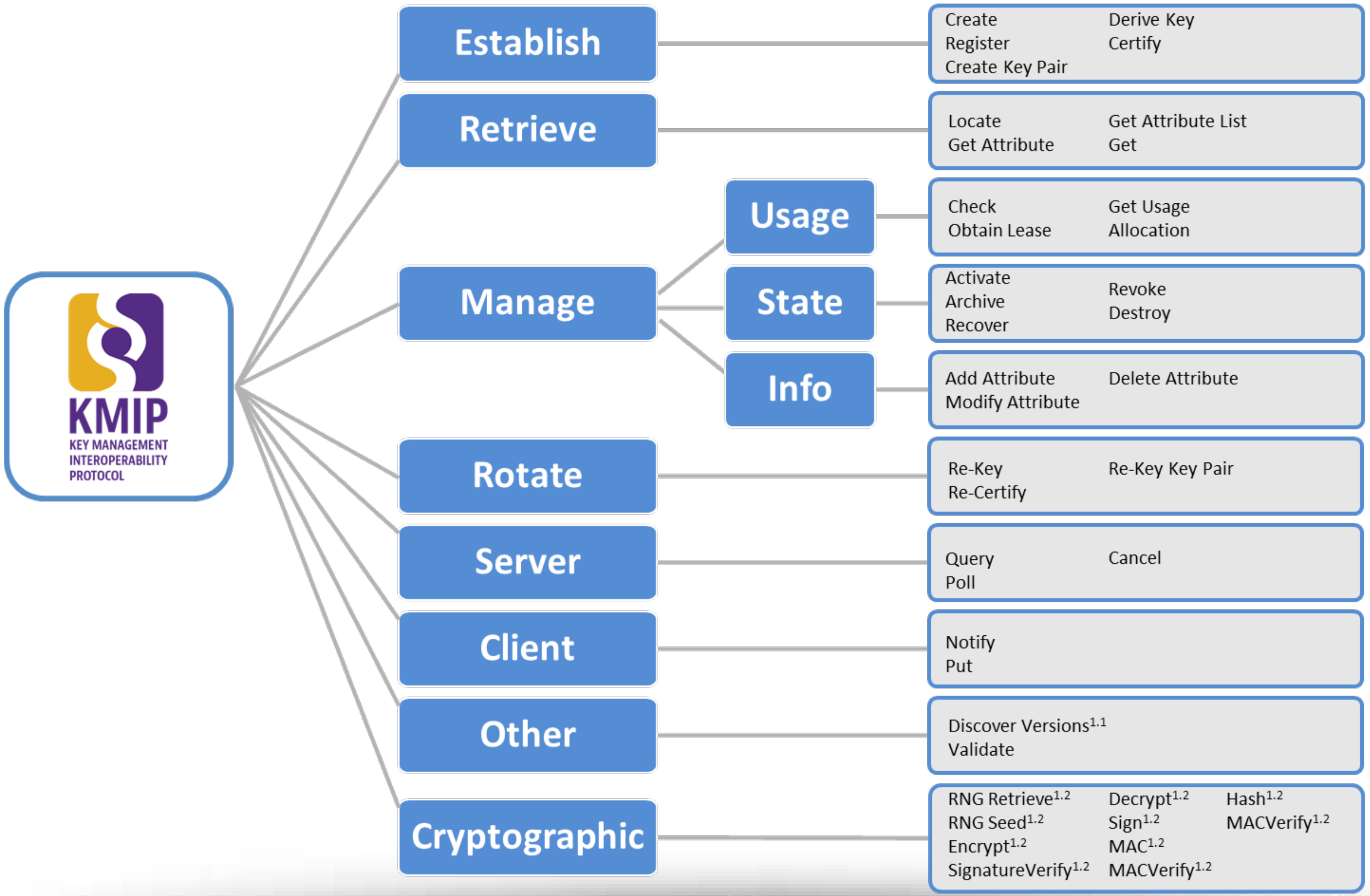
KMIP Fundamentals

- Managed Objects have an “Object Type”
  - Certificate
  - Symmetric Key
  - Public Key
  - Private Key
  - Split Key
  - Template *(Deprecated in KMIP 1.2)*
  - Secret Data
  - Opaque Object
  - PGP Key$^{1.2}$

KMIP Fundamentals

- Managed Objects have a set of “Attributes”
  - Every attribute has a string name
  - Every attribute has a type
  - May be simple types or complex types
  - Some set by server once and cannot be changed
  - Some set by client once and cannot be changed
  - Most are singleton (only one instance)
  - Server defined non-standard extensions are prefixed with “y-” in their string name
  - Client defined non-standard extensions are prefixed with “x-” in their string name

KMIP Fundamentals



OASIS KMIP vendor

IMPLEMENTATION ERRORS

Implementation Errors

- Simple
  - Invalid Padding
  - Invalid Encoding
  - Invalid Tag Values
  - Invalid Field Order
  - Invalid TLS usage
  - Missing Mandatory
  - Mandating Optional
  - Invalid sign

Implementation Errors

- Complex
  - Core concepts omitted
  - Special interpretation added
  - Conceptual confusion (Templates)
  - Unusual feature set selection
  - Assumed message sequences and content

Implementation Errors

Simple invalid encoding errors

- The specification includes clear text on encoding
- The specification includes examples of each encoding
- The KMIP 1.0 Test Cases include the hexadecimal request and response sequences
- Almost every vendor gets one or more of the encoding items wrong

Implementation Errors

9.1.1.3 Item Length

An Item Length is a 32-bit binary integer, transmitted big-endian, containing the number of bytes in the Item Value.

Data Type	Length
Structure	Varies, multiple of 8
Integer	4
Long Integer	8
Big Integer	Varies, multiple of 8
Enumeration	4
Boolean	8
Text String	Varies
Byte String	Varies
Date-Time	8
Interval	4

Actual Implementation Errors

- No padding
- Padding before rather than at end of value
- Padding missing for some types
- Padding added for types that do not require padding

If the Item Type is Structure, then the Item Length is the total length of all of the sub-items contained in the structure, including any padding. If the Item Type is Integer, Enumeration, Text String, Byte String, or Strings SHALL be padded with the minimal number of bytes following the Item Value to obtain a multiple Value.

Implementation Errors - Solution

Simple invalid encoding

- Accept that adding more specification text does not fix this issue
- Accept that adding more examples of encoding are the same as adding more specification text – they are simply either not read or not read carefully
- Accept that test cases seem to be ignored more often than they are used

Implementation Errors - Solution

Simple invalid encoding errors

- Test interoperability between implementations
  - More plug-fests
  - More interop-events
  - More tests defined in more approachable manner
  - Formal conformance testing program

i.e. more events and wider scope

Implementation Errors

Special interpretation or conceptual confusion

- Adding semantics that don’t exist – leaping beyond the spec to non-interoperable solutions
  - Using *Templates* for policy management
  - Automatically creating objects during search
  - Ignoring Password fields (accept anything)
  - Requiring Names
  - Forcing restricted set of characters in Names

Implementation Errors - Solution

Special interpretation or conceptual confusion

- Deprecated *Templates* as of KMIP 1.2
- Require explicit indication for create-when-searching if really necessary
- Adding Alternate Name and “vendor education”
- Expanding testing of Names which exceed arbitrary restrictions (spaces, punctuation, etc)
- More test cases and profiles
- Flexible interpretation in servers

Implementation Errors

Assumed message sequences and content

- Pattern matching rather than understanding
  - Ignoring most of the message content
  - Assuming fixed list of fields in fixed order for non-ordered lists
  - Assuming fixed sequence of request / response items
  - Pre-canned responses with minimal substitution
  - Ignoring protocol version information

Implementation Errors - Solution

Assumed message sequences and content

- Detect this sort of implementation
- Determine limitations of the approach
- Expand on testing to require more semantic processing rather than simple syntax
- More test cases and profiles

Guidance for key vendor issues in

KEY MANAGEMENT

Guidance

- Fundamental Requirements
  - Don’t lose the keys
    - Don’t break the device or application using keys
  - Don’t stop serving keys when they are needed
    - Don’t stop the device or application keys from working
  - Don’t give the keys to the wrong person
    - Don’t break the purpose of adding encryption by undoing the security properties

Guidance

- Context
  - Context free key management is low value
  - Anonymous keys don't allow for active security management or meaningful auditing
  - How much context can be provided
    - KMIP has no fundamental (practical) limits on attaching context and cross-relating keys

Guidance

- Clear requirements
  - What do you want for interoperability now
  - What are you likely to want in the future
  - How do your products use key management
  - How will your security administrators use key management
  - What are your target number of keys and access patterns
    - Performance radically varies between vendors

Danger signs in vendor approaches to

KEY MANAGEMENT

Danger Signs

- **Only** indication of KMIP support is in product data sheet
- Vendor-specific implementation and **no interoperability indicators** (no plug-fest, no-interop, no conformance report, no vendor-to-vendor KMIP integration claims)

Danger Signs

- Key management integrations listed without making it clear **which protocol** is being used
  - Claims of legacy protocol integrations not separated from KMIP integrations
  - Server supports KMIP; Client supports server does not mean client uses KMIP
- Capabilities not clearly separated between vendor protocol and KMIP
  - Creative marketing messages

Summary on multi-vendor

KEY MANAGEMENT

Summary

- Capability and claims vary substantially
- Verify claims – don't make assumptions
- Interoperability is only actually achieved when products work together
- Conformance testing programs provide assurance and reduce the burden of point-to-point testing

SDC

STORAGE DEVELOPER CONFERENCE

SNIA ■ SANTA CLARA, 2015

Multi-Vendor Key Management with KMIP

Tim Hudson

CTO & Technical Director

CRYPTSOFT

tjh@cryptsoft.com