SNIA DSI '16 DATA STORAGE INNOVATION CONFERENCE
Innovation in Storage Products, Services, and Solutions
June 13-15, 2016 | Marriott San Mateo | San Mateo, CA

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

KEY MANAGEMENT

Why Key Management

- Encryption is the primary means of securing stored data
- Data Encryption without adequate key management is pointless

Encryption impact on Data Protection

- Data Protection (Storage)

  Assurance that data is not corrupted, is accessible for authorized purposes only, and is in compliance with applicable requirements.

  - Storage Networking Industry Association Dictionary

- Data Protection (Security)

  The implementation of appropriate administrative, technical or physical means to guard against unauthorized intentional or accidental disclosure, modification, or destruction of data.

  - ISO/IEC 2382-1:1993

Source: Eric Hibbard – Hitachi Data Systems

Why Key Management Standards

- Standards are of limited use if not implemented properly
- Standards are also of limited use if they not widely adopted

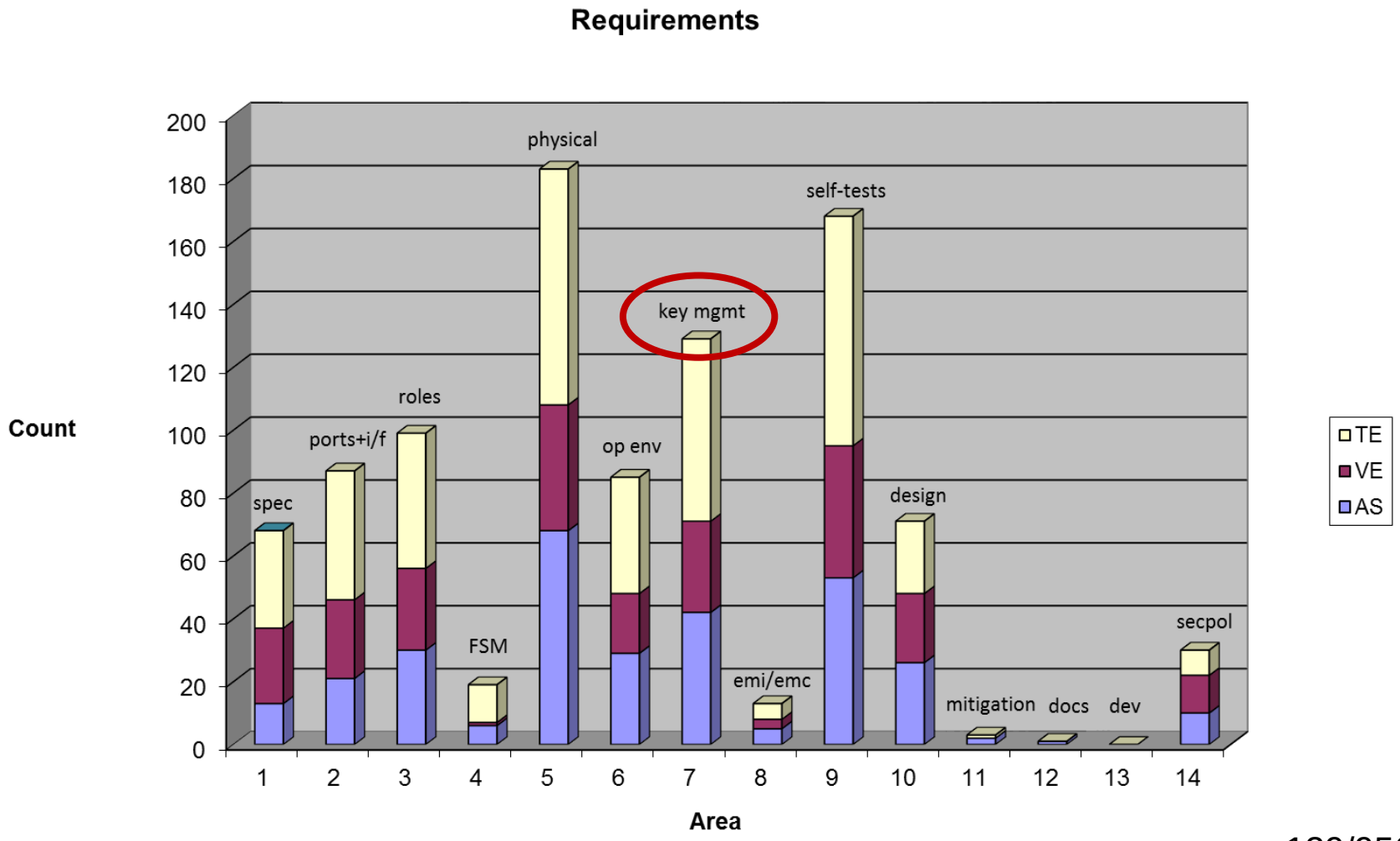
Key Management Standards

- NSA EKMS
- OASIS EKMI
- ANSI X9.24
- IEEE P1619.3
- OASIS KMIP
- IETF KEYPROV
- NIST SP 800-57
- NIST SP 800-130
- NIST SP 800-152
- ISO 11770

Specifications & Standards

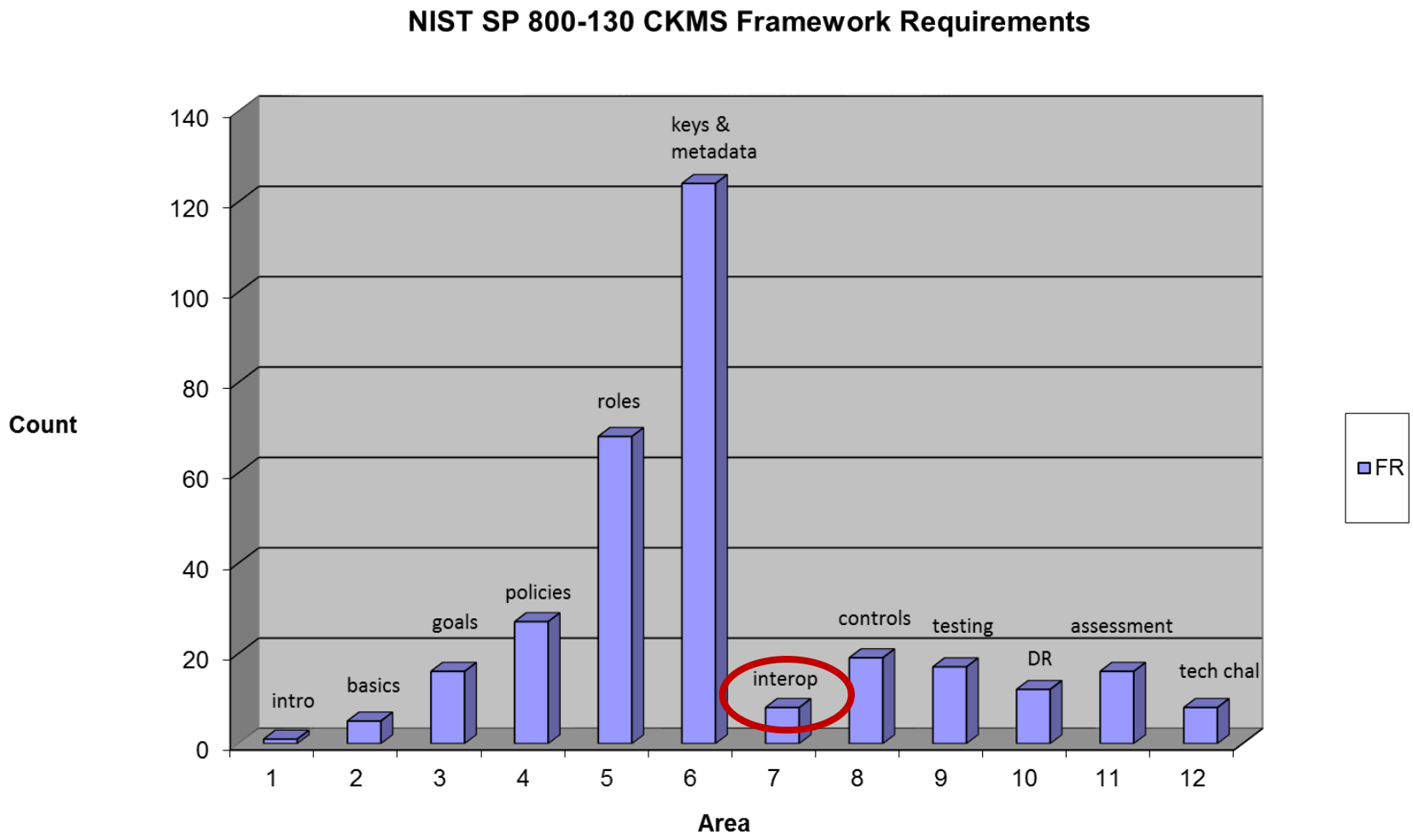
- ISO/IEC
  - ISO/IEC 11770
  - *ISO/IEC 27040, ISO/IEC 18033*
- NIST
  - NIST FIPS 140-2, NIST SP 800-57
  - NIST SP 800-130, NIST SP 800-152
  - *NIST SP 800-88*
- OASIS
  - KMIP (Key Management Interoperability Protocol)

FIPS 140-2 Key Management



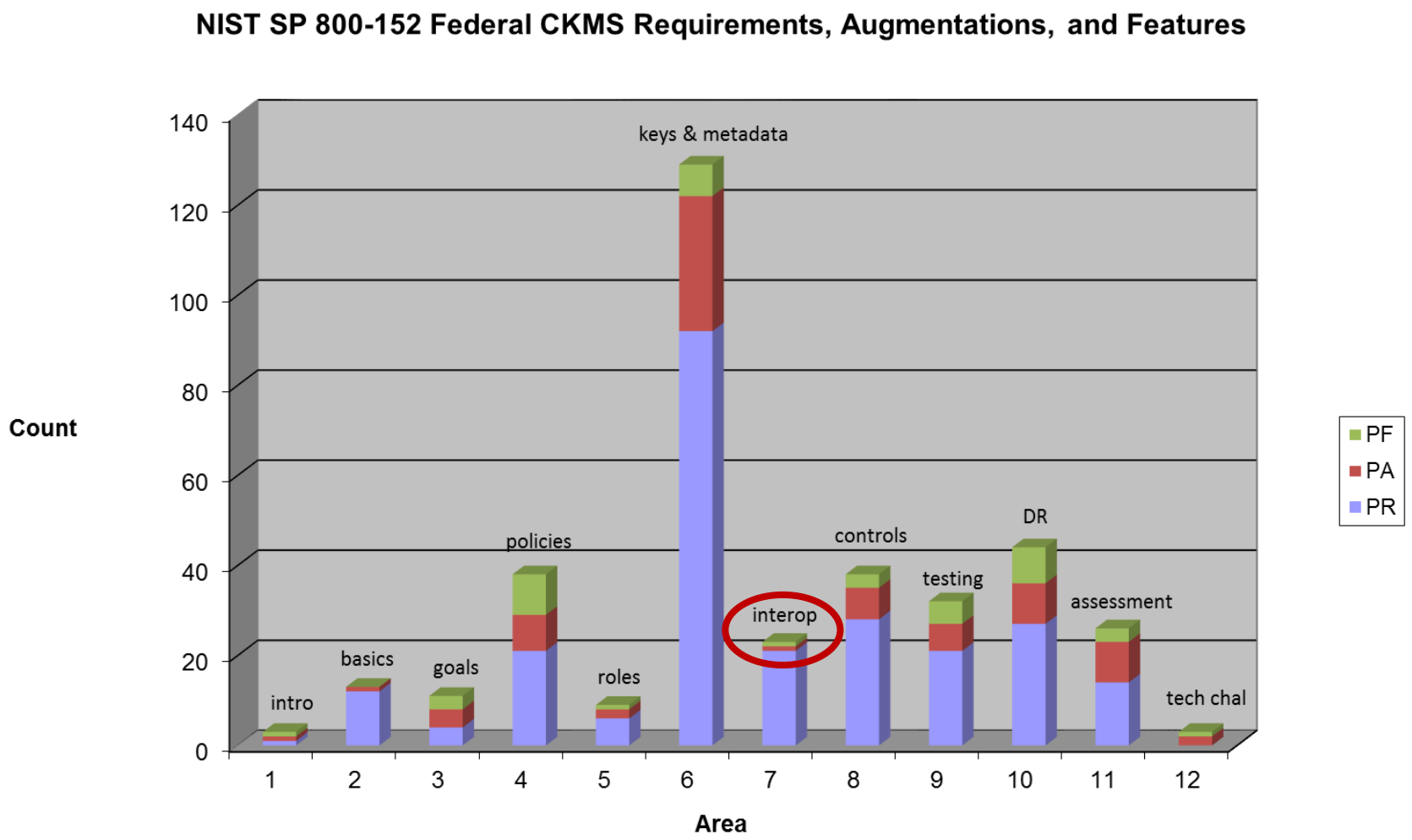
129/956

NIST SP 800-130 CKMS



8/321

NIST SP 800-152 Federal KM Profile

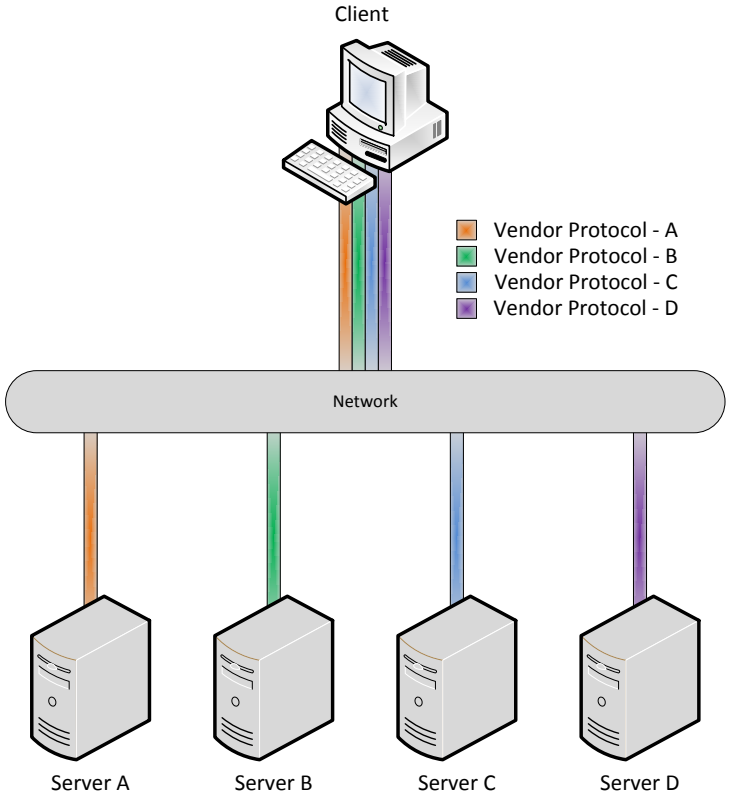


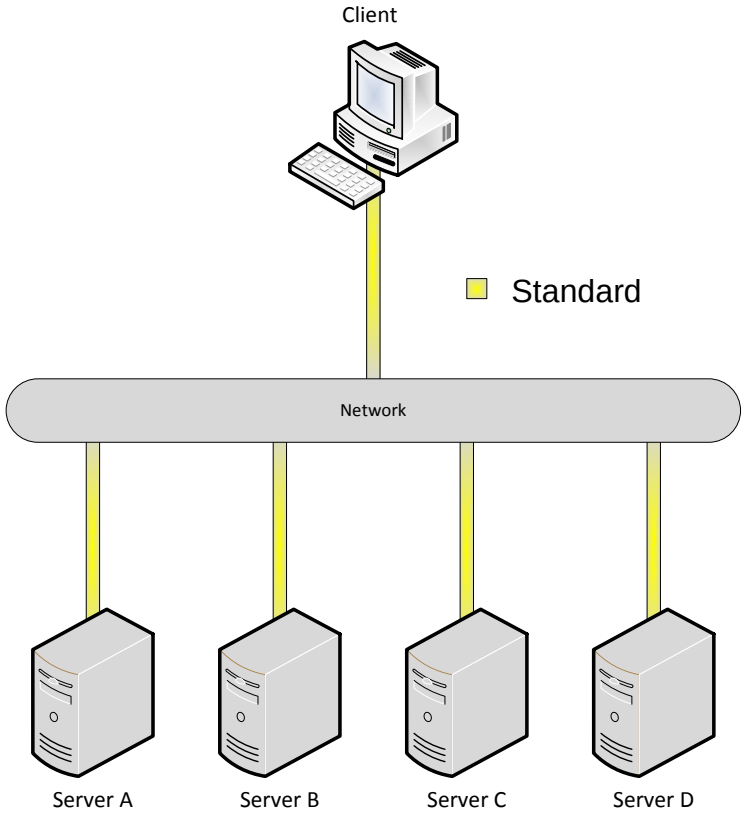
23/369

The need for multi-vendor

KEY MANAGEMENT

Multi-Vendor – Single Integration





Prior to a standard each application had to support each vendor protocol

With a standard 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
  - Must 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

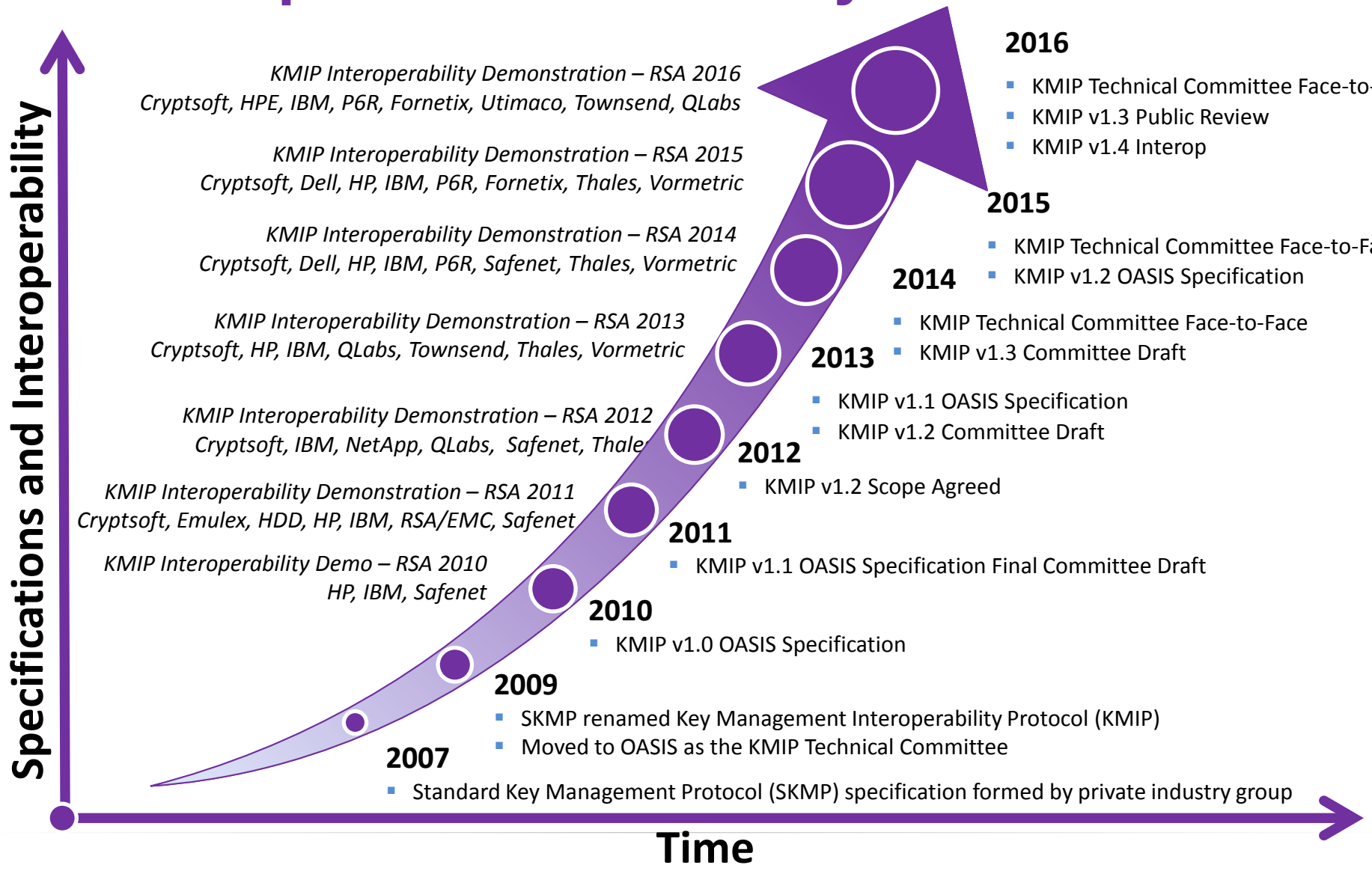
KEY MANAGEMENT

What is KMIP?

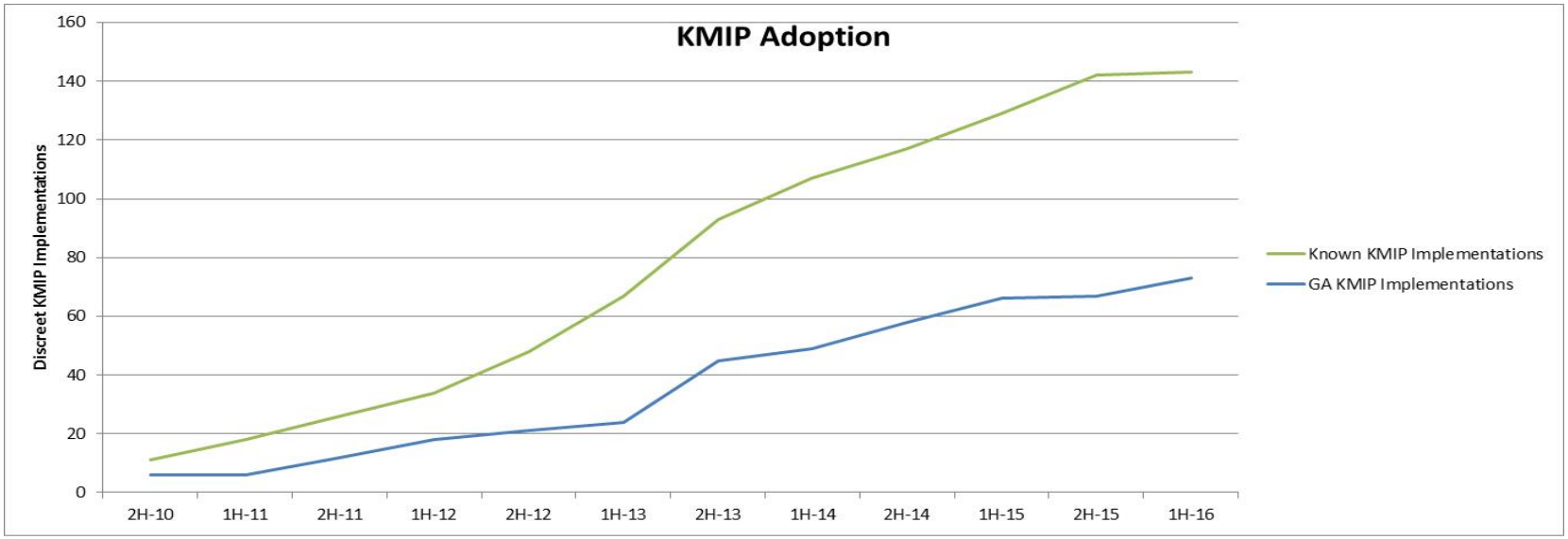
- Key Management Interoperability Protocol
  - "The OASIS KMIP TC works to define a single, comprehensive protocol for communication between encryption systems and a broad range of new and legacy enterprise applications, including email, databases, and storage devices. By removing redundant, incompatible key management processes, KMIP will provide better data security while at the same time reducing expenditures on multiple products." - https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=kmip
  - A protocol for enterprise management of "stuff"

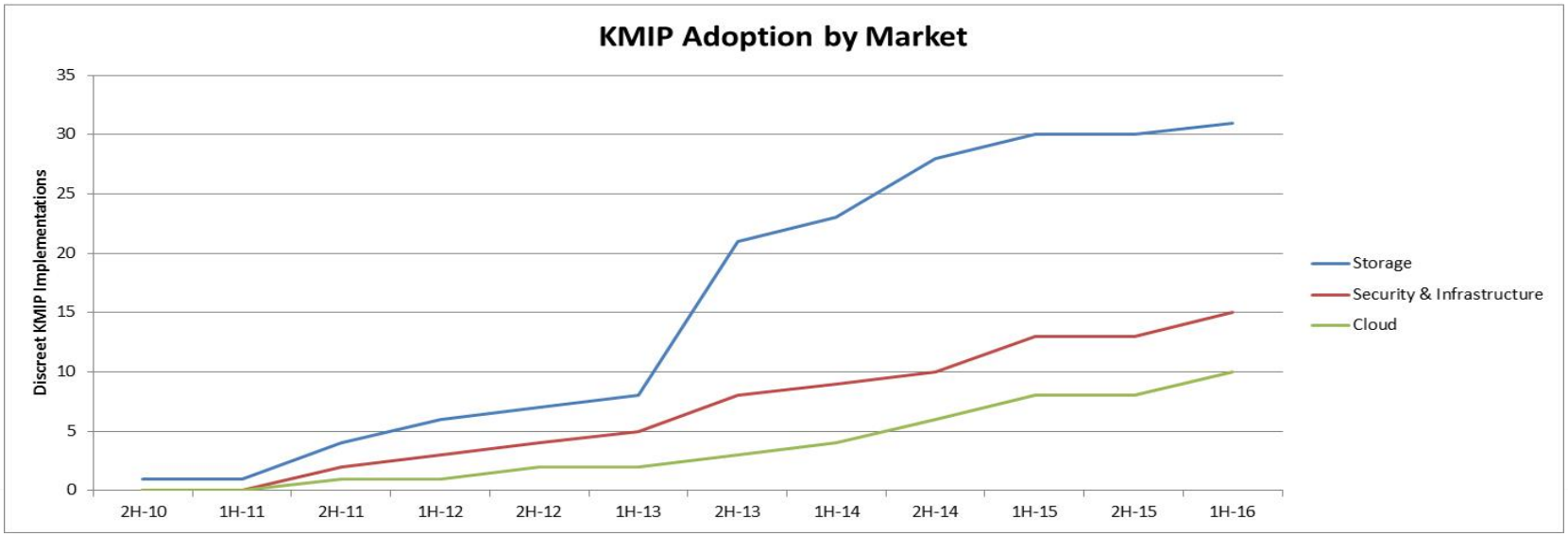
- OASIS KMIP TC Membership (foundational and sponsor)
  - Cryptsoft, Dell, EMC, Fornetix, Futurex, Hancom Secure, Hewlett Packard Enterprise, IBM, NetApp, Oracle, SafeNet, Symantec, VMware, Vormetric

KMIP Specification History



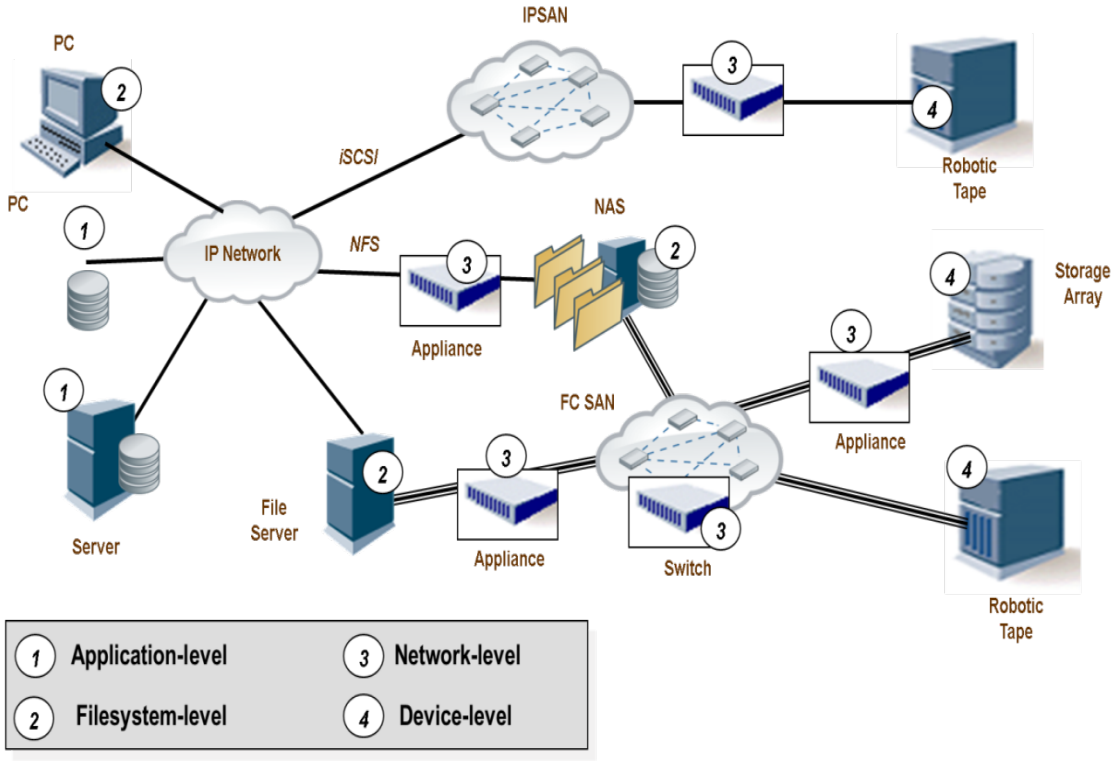
Multi-Vendor – How many products





KMIP – Adoption (Storage)

- **KMIP is present in the following:**
  - Device-level
    - Disk arrays
    - Tape libraries
    - Virtual tape libraries
    - Flash storage arrays
    - Storage controllers
    - Storage operating systems
  - Network-level
    - Encrypting switches
  - File/Object-level
    - NAS appliances
  - Application-level



Source: ISO/IEC 27040 - Information technology - Security techniques - Storage security

Multi-Vendor – Who and Where

Storage	Infrastructure and Security	Cloud
▪ Disk Arrays, Flash Storage Arrays, NAS Appliances ▪ Tape Libraries, Virtual Tape Libraries ▪ Encrypting Switches ▪ Storage Key Managers ▪ Storage Controllers ▪ Storage Operating Systems	▪ Key Managers ▪ Hardware security modules ▪ Encryption Gateways ▪ Virtualization Managers ▪ Virtual Storage Controllers ▪ Network Computing Appliances	▪ Key Managers ▪ Compliance Platforms ▪ Information Managers ▪ Enterprise Gateways and Security ▪ Enterprise Authentication ▪ Endpoint Security

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)

OASIS KMIP

SPECIFICATION

KMIP Specification History

PR = Public Review
CS = Committee Specification
OS = OASIS Standard

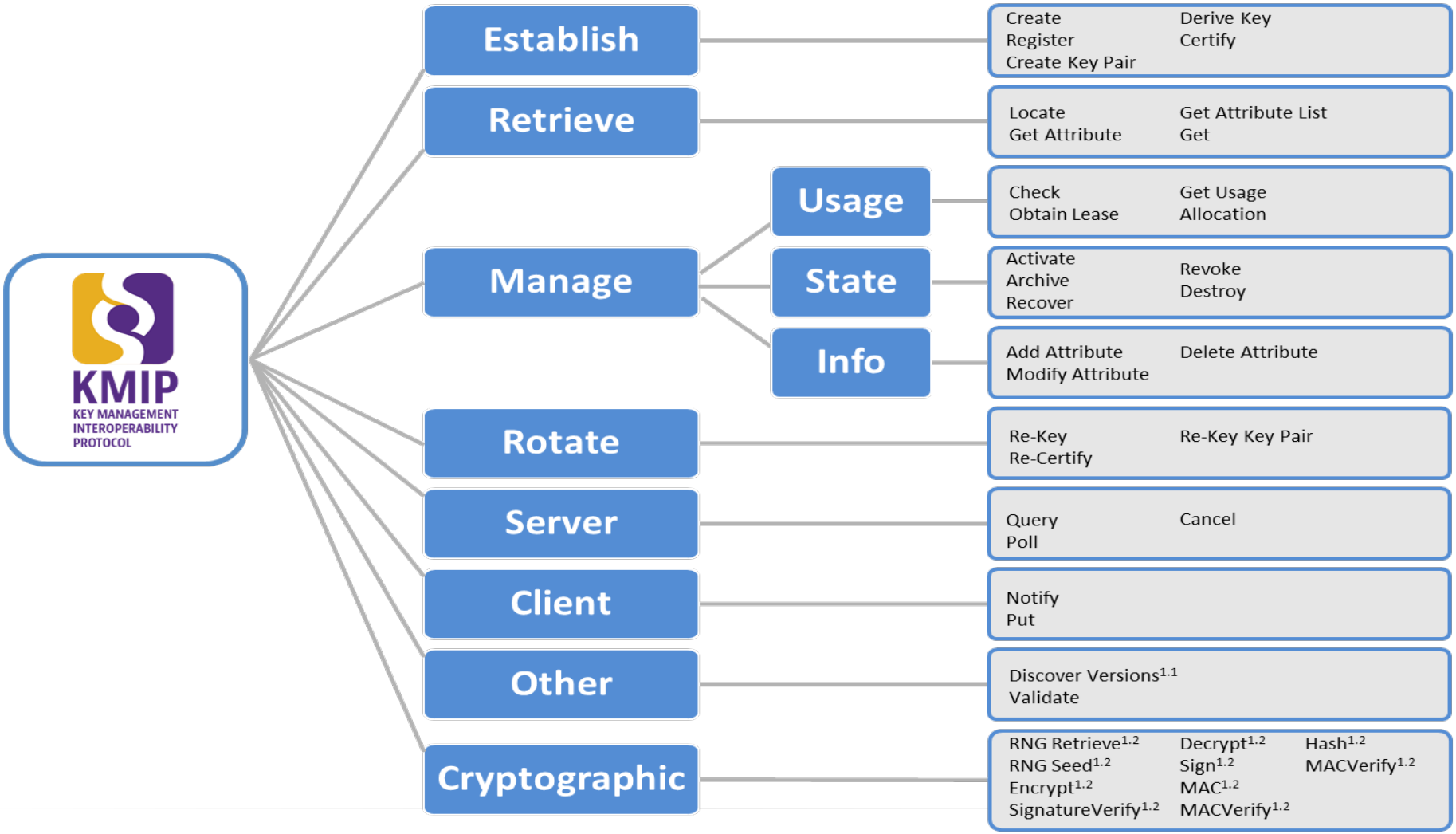
- OASIS KMIP 1.0 – PR Nov 2009, CS Jun 2010, OS Oct 2010
  - Specification 105 pages
  - Profiles 16 pages
  - *Usage Guide* *44 pages*
  - *Use Cases (Test Cases)* *168 pages*
- OASIS KMIP 1.1 – PR Jan 2012, CS Jul 2012, OS Jan 2013
  - Specification 164 pages +56%
  - Profiles 39 pages +143%
  - *Usage Guide* *63 pages* *+43%*
  - *Test Cases* *513 pages* *+205%*
- OASIS KMIP 1.2 – PR Jan 2014, CS Nov 2014, OS 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*

OASIS KMIP - Protocol Concepts

Core Concepts

- Base Objects
  - Protocol building blocks and parameter encoding
- Managed Objects
  - Core concepts managed by KMIP
  - Cryptographic Managed Objects (objects with key material)
- Attributes
  - Details related to or about a managed object
- Client-to-Server Operations
  - Operations clients can send in requests to servers
- Server-to-Client Operations
  - Operations servers can send in requests to clients
- Message Contents and Message Formats
  - Request and Response protocol messages
- Message Encoding
  - Binary Tag-Type-Length-Value
- Authentication
  - See Profiles (Client Certificates)
- Transport
  - See Profiles (TLSv1.0 or TLSv1.2)

KMIP Fundamentals - Operations



26

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

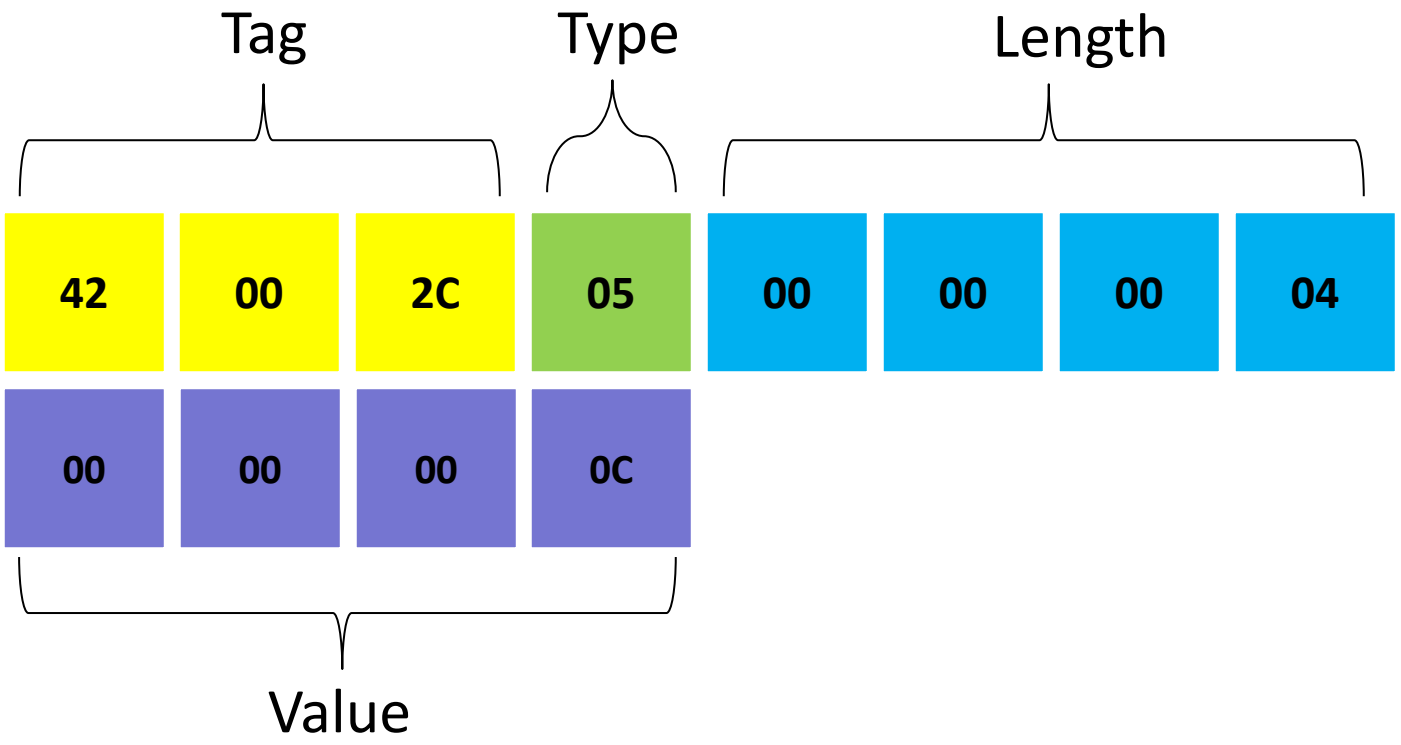
- Managed Objects have an “Object Type”
  - Certificate
  - Symmetric Key
  - Public Key
  - Private Key
  - Split Key
  - Template *(Deprecated in KMIP 1.2)*
  - Secret Data
  - Opaque Object
  - PGP Key[1.2]

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 – Message Encoding

- Binary Tag-Type-Length-Value format
- Optional JSON and XML encoding in KMIP[1.2]



Cryptographic Usage Mask = Encrypt | Decrypt

KMIP Fundamentals - TTLV

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								

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>
```

KMIP Fundamentals - JSON

```
{"tag":"RequestMessage", "value":[
  {"tag":"RequestHeader", "value":[
    {"tag":"ProtocolVersion", "value":[
      {"tag":"ProtocolVersionMajor", "type":"Integer", "value":"0x00000001"},
      {"tag":"ProtocolVersionMinor", "type":"Integer", "value":"0x00000000"}
    ]},
    {"tag":"BatchCount", "type":"Integer", "value":"0x00000001"}
  ]},
  {"tag":"BatchItem", "value":[
    {"tag":"Operation", "type":"Enumeration", "value":"Create"},
    {"tag":"RequestPayload", "value":[
      {"tag":"ObjectType", "type":"Enumeration", "value":"SymmetricKey"},
      {"tag":"TemplateAttribute", "value":[
        {"tag":"Attribute", "value":[
          {"tag":"AttributeName", "type":"TextString", "value":"Cryptographic Algorithm"},
          {"tag":"AttributeValue", "type":"Enumeration", "value":"AES"}
        ]},
        {"tag":"Attribute", "value":[
          {"tag":"AttributeName", "type":"TextString", "value":"Cryptographic Length"},
          {"tag":"AttributeValue", "type":"Integer", "value":"0x00000080"}
        ]},
        {"tag":"Attribute", "value":[
          {"tag":"AttributeName", "type":"TextString", "value":"Cryptographic Usage Mask"},
          {"tag":"AttributeValue", "type":"Integer", "value":"Decrypt|Encrypt"}
        ]}
      ]}
    ]}
  ]}
]}
```

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

The importance of vendor-independent

CONFORMANCE TESTING

KMIP Conformance Testing - Intent

- The SNIA – Storage Security Industry Forum (SSIF) launched the program in response to market demand
- The program enables organizations to shortlist vendor KMIP solutions based on support for specific usage scenarios and interoperability
- Enables organizations to verify vendor claims
- Value provided by a truly independent test team

KMIP Conformance Testing - Profiles

- The KMIP Technical Committee defines Profiles
  - Normative documents specify minimum set of supported functionality
  - Contain expected requests and responses
  - Cover a range of deployment scenarios

KMIP Profiles

- Advanced Cryptographic Client & Server[1.2]
- Advanced Symmetric Key Foundry Client & Server
- Asymmetric Key Lifecycle Client & Server
- Baseline Client & Server Basic
- Baseline Client & Server TLSv1_2
- Basic Cryptographic Client & Server[1.2]
- Basic Symmetric Key Foundry Client & Server
- HTTPS, JSON, XML Client & Server
- Intermediate Symmetric Key Foundry Client & Server
- Opaque Managed Object Store Client & Server
- RNG Cryptographic Client & Server[1.2]
- Storage Array With SED Client & Server
- Suite-B MinLOS_128 Client & Server
- Suite-B MinLOS_192 Client & Server
- Symmetric Key Lifecycle Client & Server
- Tape Library Client & Server
- Complete Server

54

KMIP Conformance Testing - Method

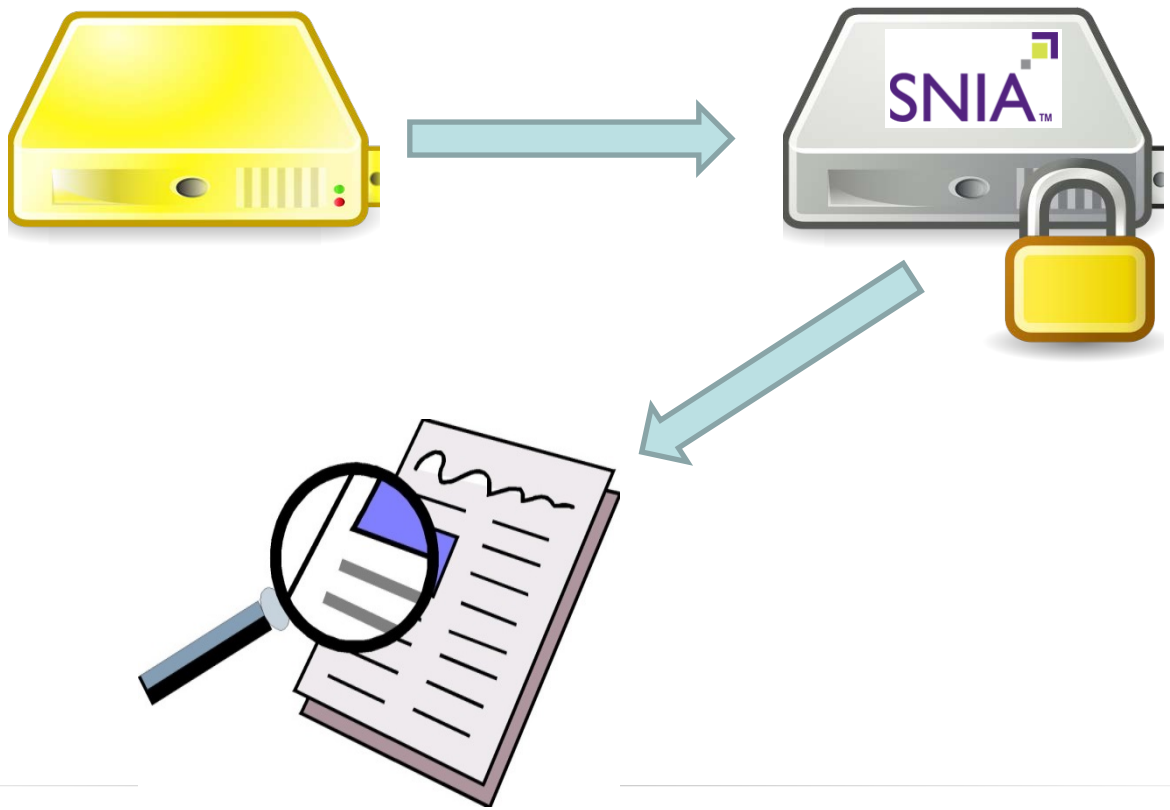
- Implementations are made available to the test team
  - Implementations may be tested onsite or remotely
- Test team operates under the SSIF's direction but testing information is kept completely confidential
- Test Report is provided to the customer

55

KMIP Conformance Testing - Client

Customer Storage Product (KMIP Client)

SSIF Test Infrastructure



56

KMIP Profile – Example (SASED)

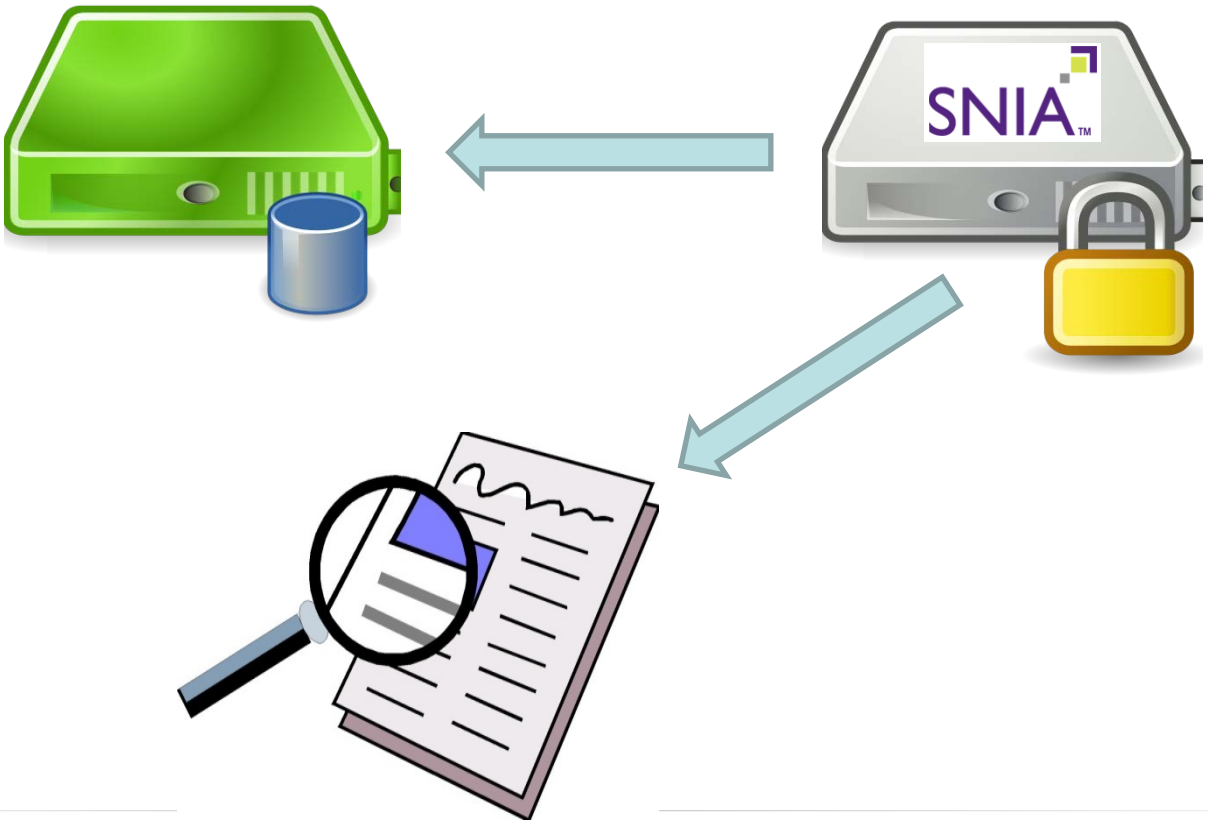
```
      # TIME 0
0001  <RequestMessage>
0002    <RequestHeader>
0003      <ProtocolVersion>
0004        <ProtocolVersionMajor type="Integer" value="1"/>
0005        <ProtocolVersionMinor type="Integer" value="0"/>
0006      </ProtocolVersion>
0007      <BatchCount type="Integer" value="1"/>
0008    </RequestHeader>
0009    <BatchItem>
0010      <Operation type="Enumeration" value="Query"/>
0011      <RequestPayload>
0012        <QueryFunction type="Enumeration" va
0013        <QueryFunction type="Enumeration" va
0014        <QueryFunction type="Enumeration"
      value="QueryServerInformation"/>
0015      </RequestPayload>
0016    </BatchItem>
0017  </RequestMessage>
```

```
0018  <ResponseMessage>
0019    <ResponseHeader>
0020      <ProtocolVersion>
0021        <ProtocolVersionMajor type="Integer" value="1"/>
0022        <ProtocolVersionMinor type="Integer" value="0"/>
0023      </ProtocolVersion>
0024      <TimeStamp type="DateTime" value="2013-04-25T16:53:03+00:00"/>
0025      <BatchCount type="Integer" value="1"/>
0026    </ResponseHeader>
0027    <BatchItem>
0028      <Operation type="Enumeration" value="Query"/>
0029      <ResultStatus type="Enumeration" value="Success"/>
0030      <ResponsePayload>
0031        <Operation type="Enumeration" value="Query"/>
0032        <Operation type="Enumeration" value="Locate"/>
0033        <Operation type="Enumeration" value="Destroy"/>
0034        <Operation type="Enumeration" value="Get"/>
0035        <Operation type="Enumeration" value="Register"/>
0036        <Operation type="Enumeration" value="GetAttributes"/>
0037        <Operation type="Enumeration" value="GetAttributeList"/>
0038        <Operation type="Enumeration" value="AddAttribute"/>
0039        <ObjectType type="Enumeration" value="SecretData"/>
0040        <ObjectType type="Enumeration" value="Template"/>
0041        <VendorIdentification type="TextString" value="server-vendor.com"/>
            <ServerInformation>
0042        </ServerInformation>
0043      </ResponsePayload>
0044    </BatchItem>
0045  </ResponseMessage>
0046
```

KMIP Conformance Testing - Server

Customer Storage Server (KMIP Server)

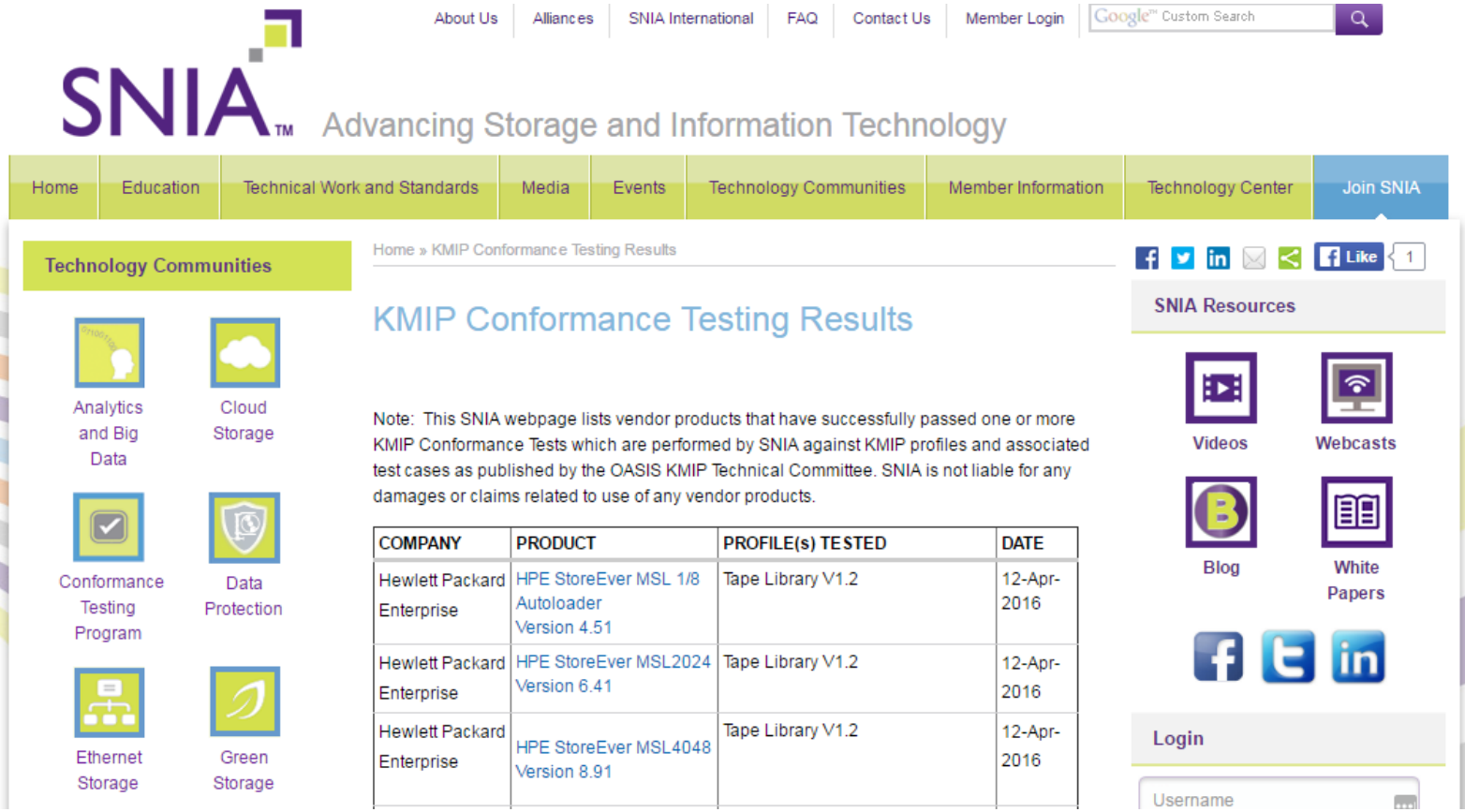
SSIF Test Infrastructure



KMIP Conformance Testing - Results

- Results remain entirely confidential to customer and conformance test team until results are published
- Test results are published with customer's permission by SNIA SSIF
- Only the successful results (supported profiles) appear on the results page
  - Failures and/or non-supported profiles are not stated

KMIP Conformance Testing - Results



About Us | Alliances | SNIA International | FAQ | Contact Us | Member Login | Google™ Custom Search

SNIA™ Advancing Storage and Information Technology

Home | Education | Technical Work and Standards | Media | Events | Technology Communities | Member Information | Technology Center | Join SNIA

Technology Communities

- Analytics and Big Data
- Cloud Storage
- Conformance Testing Program
- Data Protection
- Ethernet Storage
- Green Storage

Home » KMIP Conformance Testing Results

KMIP Conformance Testing Results

Note: This SNIA webpage lists vendor products that have successfully passed one or more KMIP Conformance Tests which are performed by SNIA against KMIP profiles and associated test cases as published by the OASIS KMIP Technical Committee. SNIA is not liable for any damages or claims related to use of any vendor products.

COMPANY	PRODUCT	PROFILE(s) TESTED	DATE
Hewlett Packard Enterprise	HPE StoreEver MSL 1/8 Autoloader Version 4.51	Tape Library V1.2	12-Apr-2016
Hewlett Packard Enterprise	HPE StoreEver MSL2024 Version 6.41	Tape Library V1.2	12-Apr-2016
Hewlett Packard Enterprise	HPE StoreEver MSL4048 Version 8.91	Tape Library V1.2	12-Apr-2016

Like 1

SNIA Resources

- Videos
- Webcasts
- Blog
- White Papers

Login

Username

Snapshot taken from : http://www.snia.org/forums/SSIF/kmip/results

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

SNIA DSI '16 DATA STORAGE INNOVATION CONFERENCE
Innovation in Storage Products, Services, and Solutions
June 13-15, 2016 | Marriott San Mateo | San Mateo, CA

Multi-Vendor Key Management with KMIP

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