STORAGE DEVELOPER CONFERENCE
SDC 21
BY Developers FOR Developers
Virtual Conference
September 28-29, 2021
A SNIA Event

Fine Grain Encryption Control For Enterprise Applications

With TCG's Per-I/O Encryption Key Selection

Festus Hategekimana, Intel Corporation
Frederick Knight, NetApp

Agenda

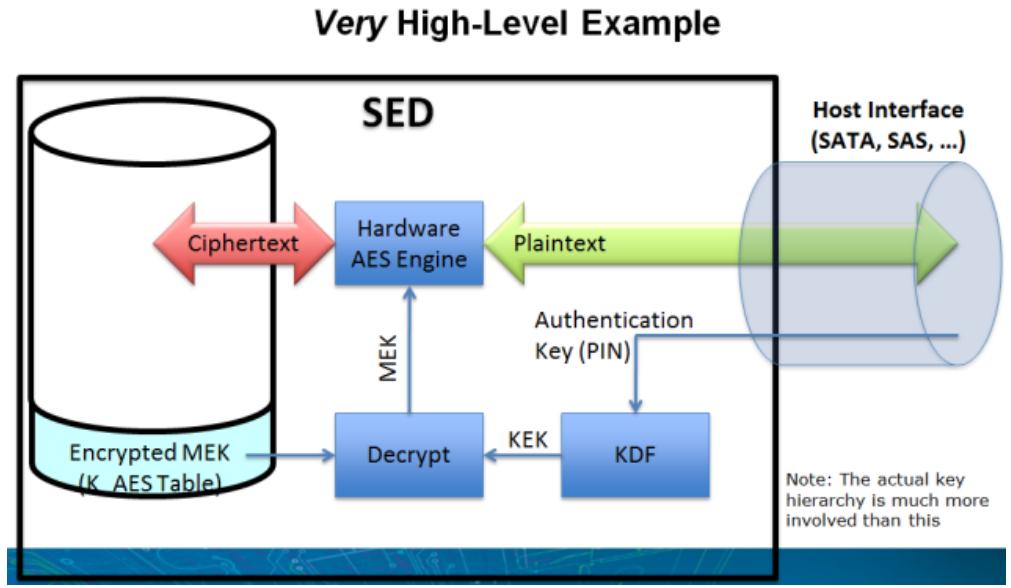
- Data at Rest Protection - Background
- Key Per I/O Overview
- Key Per I/O SSC And I/O Architectures Interactions

Key Per I/O And Data At Rest Protection

Festus Hategekimana, Intel Corporation

Background On Data At Rest Protection

Data At Rest Protection



Properties

- Encrypt all user accessible data all the time, at interface speeds
- Keys generated & stored in NVM by the storage device
- Media Encryption Key (MEK) associated with contiguous LBA ranges or Namespaces
- Opal/Enterprise SSC* deliver passwords to drive in the clear (when not using Trusted Computing Group (TCG)* - Secure Messaging)

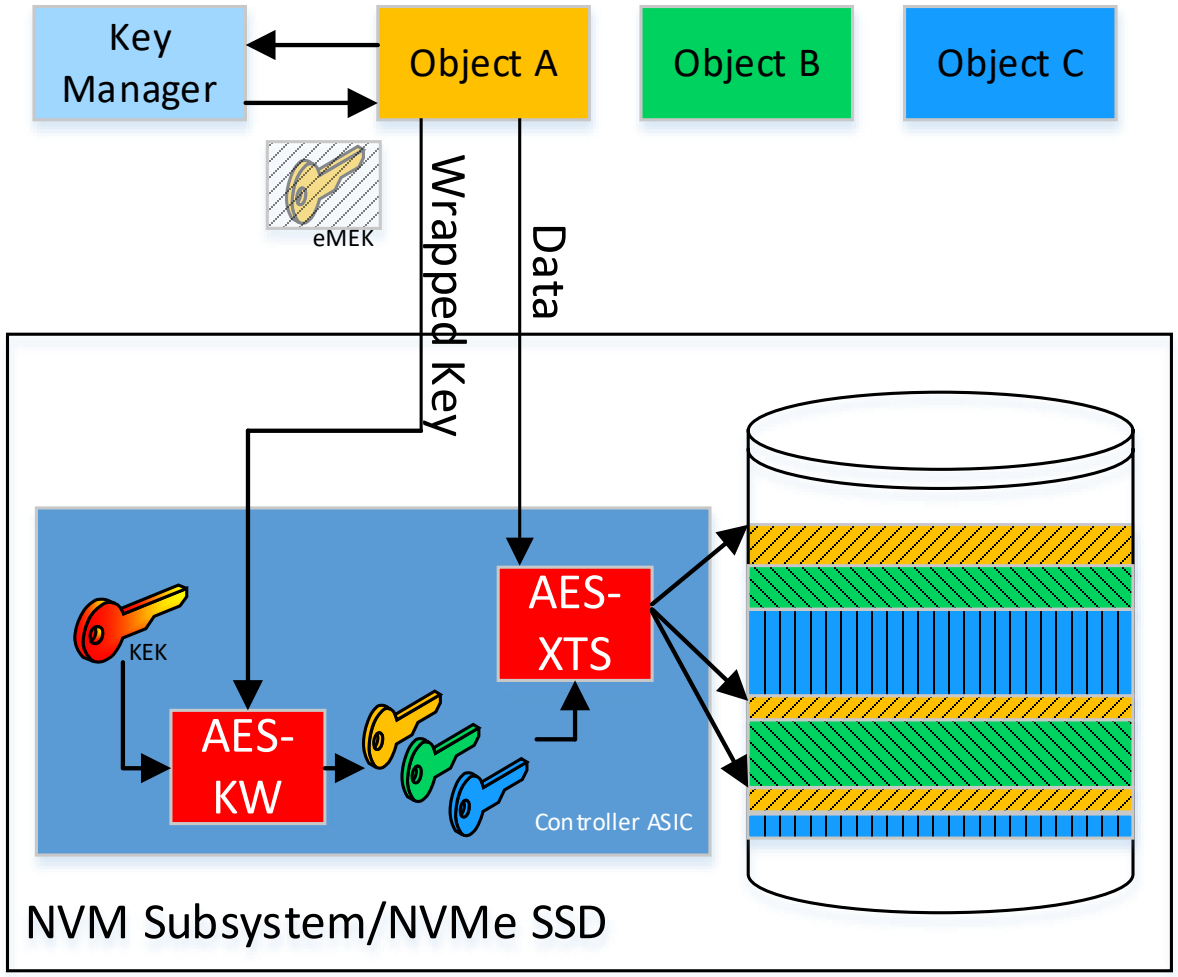
*Other names and brands may be claimed as the property of others.

Can we do better?

Desired properties:

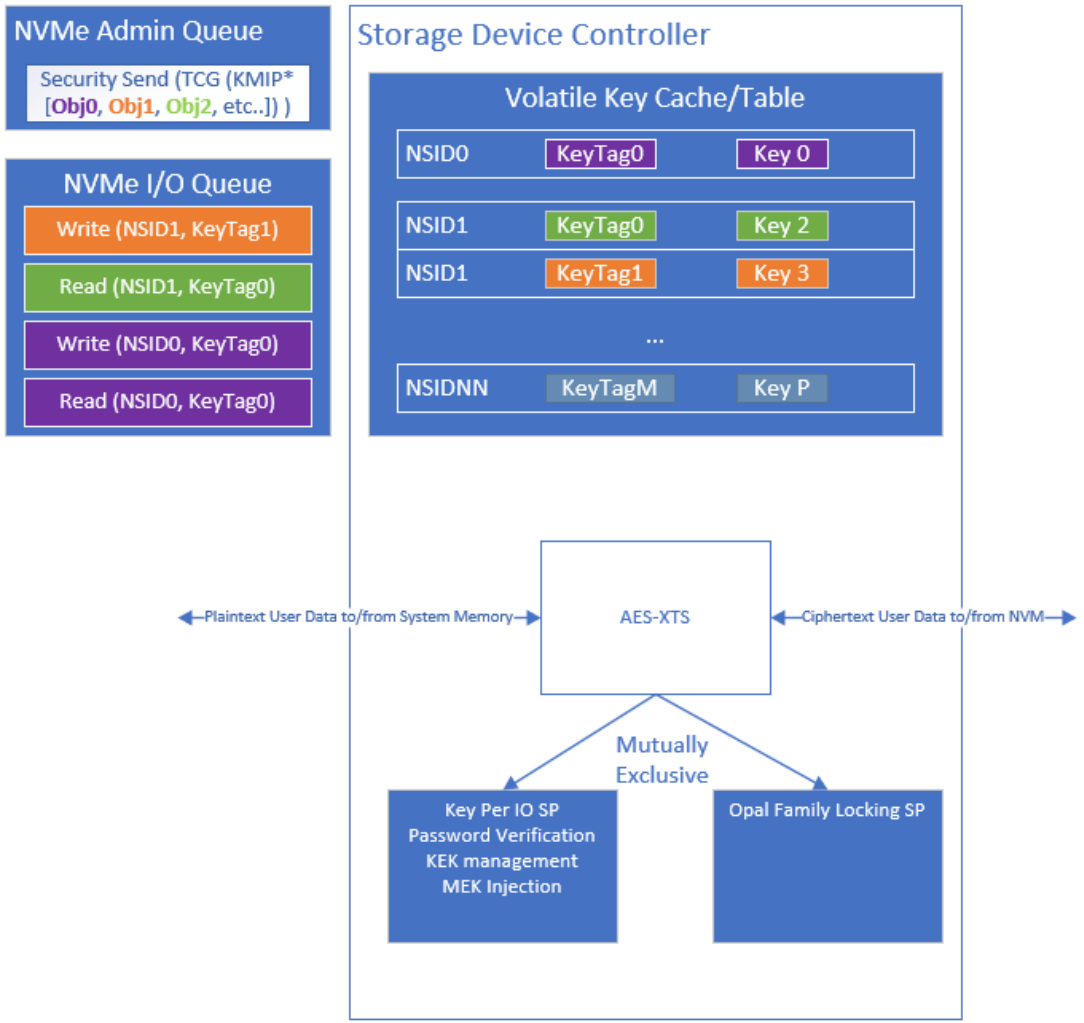
1. Select an encryption key for each I/O to a Storage Device?
   - Associate encryption domains with higher-level objects (abstractions) than drives or volumes.
   - Crypto erase individual higher-level objects
   - Easier to support European Union's General Data Protection Regulations' "Right to be forgotten"
2. Externally manage Media Encryption keys?
   - Centralized key management infrastructure, consistent key policies
   - High assurance key generation and control, e.g., master keys in HSM (Hardware Security Module)
3. Ensure that a Storage Device with no power has no encryption keys?
   - Shorter physical drive loss/theft discussion with security auditor
   - Easier decommissioning process

Key Per I/O Overview



- Encrypted Media Encryption Keys are injected into Self Encrypting Drive key cache and assigned a “Key Tag” by SW
- Subsequent I/O can use the “Key Tag” to encrypt/decrypt data to/from the storage device in a non-contiguous fashion
- Media Encryption Keys (MEKs) are encrypted (wrapped) by a Key Encryption Key (KEK)
- Media Encryption Keys (MEKs) are not stored in the NVM of the drive and are lost on power loss
- Crypto erase accomplished by deleting the MEK from the Key Manager and the SSD or by sanitizing entire SSD

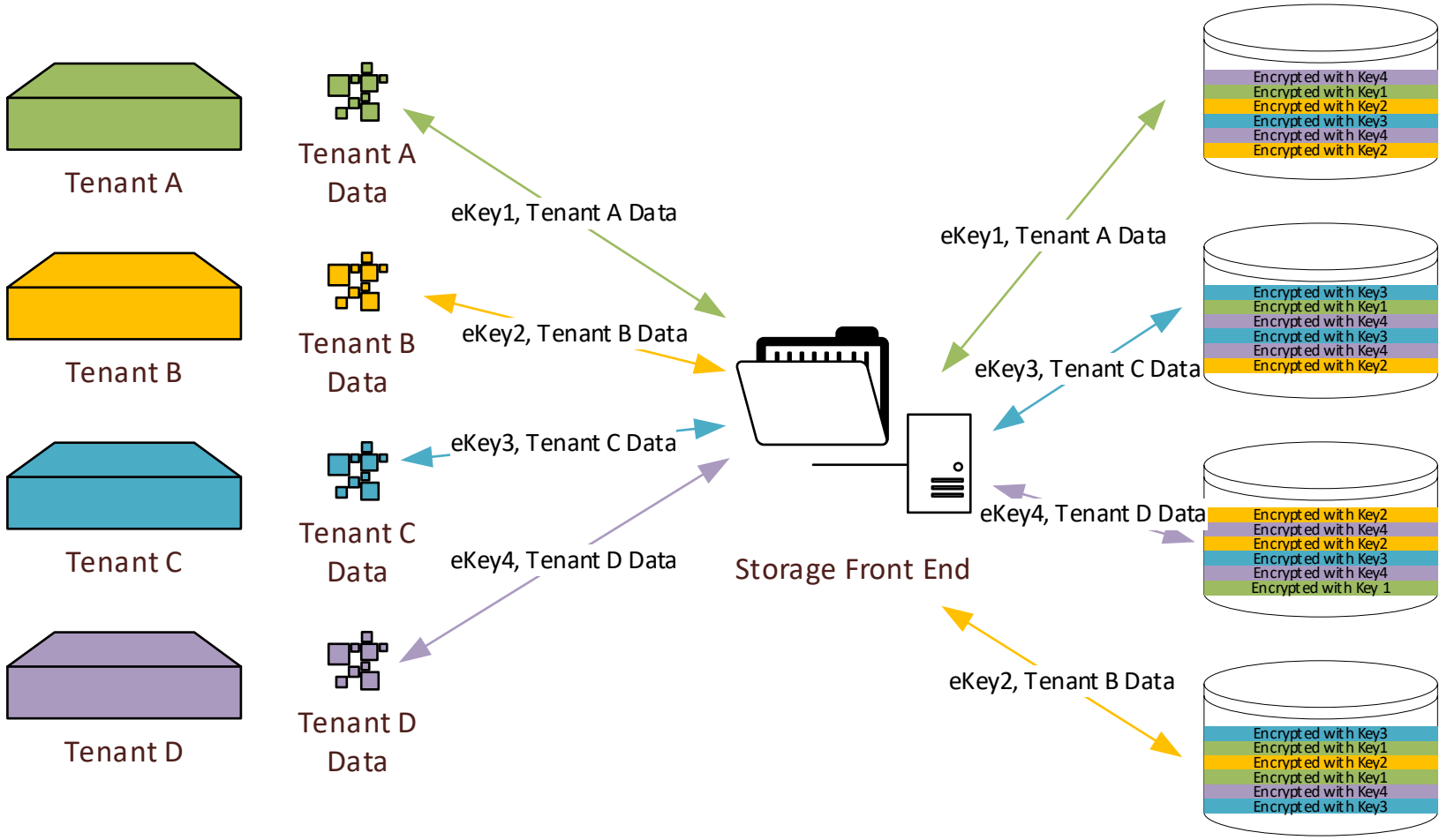
Key Per I/O Architectural Elements



- Encrypted Media Encryption Keys (eMEKs) and their wrapping Key Encryption Keys (KEKs) are injected into the storage device via the Security Send & Receive
  - Specification in progress within the TCG SWG*
- OASIS KMIP* for specifying Key data and its transportation over Security Send & Receive
  - Specification engagement in progress between TCG SWG & OASIS KMIP*
- Subsequent I/O can then use the “Key Tag”, a newly defined field in I/O commands, to specify the key that the device uses to encrypt/decrypt data to/from the storage device
  - Specification work in progress within NVMe*

*Other names and brands may be claimed as the property of others.

Data At Rest Tenant Isolation with Key Per I/O



Key Per I/O SSC And I/O Architectures Interactions

Frederick Knight, NetApp

KPIO Discovery

Host Detection of KPIO

- Number of Key Tags supported
- Granularity and alignment of operations
- NVMe Identify command
  - Per namespace
- TCG Discovery (Security Send and Security Receive)
  - Authenticate
  - Security Receive (Level 0 Discovery)
  - Discovery security characteristics

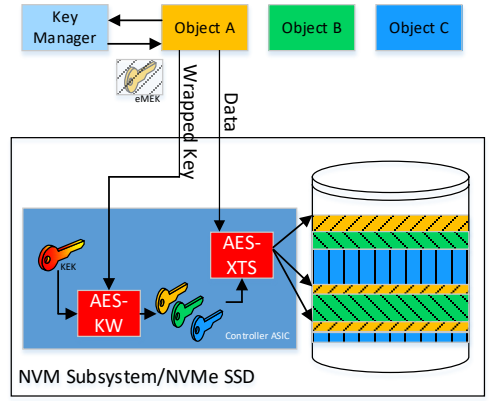
KPIO Configuration

Load the Device Key Cache

- Associate a Key Tag with a MEK
  - Per namespace – loaded using Security Send command

Key Tag	MEK (256 bit example)
1	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF
2	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE0
100	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE1
101	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE2
103	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE3
200	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE4
217	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE5

KPIO Configuration



Load New Keys

- Associate a Key Tag with a different MEK
  - Per namespace – loaded using Security Send command

Key Tag	MEK (256 bit example)
1	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF
2	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE0
100	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE6
1010	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE7
1030	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE8
2000	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE9
2170	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEA

KPIO Usage

I/O Command Usage

- Compare
- Copy
- Verify
- Read
- Write
- Write Zeroes
- Zone Append

- A field in each command to specify the Key Tag value to use for that individual I/O
- An indicator that a Key Tag is present

Key Tag	MEK (256 bit example)
1	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF
2	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE0
100	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE6
1010	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE7
1030	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE8
2000	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE9
2170	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEA

KPIO Example Commands

- WRITE (LBA=100, LEN=8, flag=1, keytag=1)

MEK = 0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF

- WRITE (LBA=200, LEN=16, flag=1, keytag=100)

MEK = 0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE6

- READ (LBA=100, LEN=8, flag=1, keytag=1)
  - Gets your data back
- READ (LBA=200, LEN=16, flag=1, keytag=1)
  - Gets error or bogus data
- READ (LBA=200, LEN=16, flag=1, keytag=100)
  - Gets your data back

Key Tag	MEK (256 bit example)
1	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF
2	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE0
100	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE6
1010	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE7
1030	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE8
2000	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE9
2170	0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEA

KPIO Impact in TCG

Security Send / Security Receive Commands

- Authentication
- Discovery
- Key Loading method (Establish Key Tag to MEK association)
- Key Remove method (Remove Key Tag to MEK association)
- Key Replacement method (Replace MEK for a Key Tag)
- Securely Purge Key Cache
- Define encryption / decryption algorithms that can be supported

KPIO Impact On Hosts

Host Responsibilities to use KPIO

- Hosts must manage the full life cycle of the Keys
- Host is responsible for the correctness of the MEK injection / key tag association and use of the correct key tag for each I/O command
- Host is responsible for preventing incorrect key tag use
  - Key tag associations change during operation – such as key tag cache size smaller than key tag needed usage
  - Using the key tag associated with the correct MEK
- Host must handle errors for improper use of key tags
  - Invalid key tag value (out of range), or a key tag with no associated MEK
  - Trying to use a key tag before injection is complete or after removal

KPIO Specification Status

Current Key Topics in Progress

- Details of the MEK / KEK loading process
  - TCG specific methods
  - KMIP based methods
  - Other methods?
- Incorrect MEK detection capability
  - Incorrect MEK should not look like a Media Error
  - Does incorrect MEK just return “bogus” data
  - UUID association
- Still a work in Process

- Work at NVMe is nearly complete
- TCG work is continuing
- Join Security BoF @ 3:00PM today
- Come join us at TCG to continue the discussions !!!!

KPIO For Other IO Architectures

What about SCSI and/or SATA

- The same TCG architecture is used by SCSI and SATA
- But completely new I/O commands would be required
  - Such as 32-byte CDBs for SCSI (to carry the Key Tag value)

- NO interest being shown to undertake such an effort

KPIO Key Takeaways

- The KPIO SSC is being defined such that an SD that claims TCG Opal SSC compatibility could be a KPIO SSC.
- Intended to protect confidentiality of data at rest from unauthorized access once it leaves the owner's control.
- Creating a fine-grained approach to enhance SED technology to better support multi-tenancy usage models.
- Standards based designs for multi-vendor interoperability.

Please take a moment to rate this session.

Your feedback is important to us.

