

TCG SSC: Key Per IO

USENIX Vault ‘20

Frederick Knight, Sridhar Balasubramanian
NetApp, Inc.,
February 2020

© 2020 NetApp, Inc. All rights reserved.

Agenda Slide



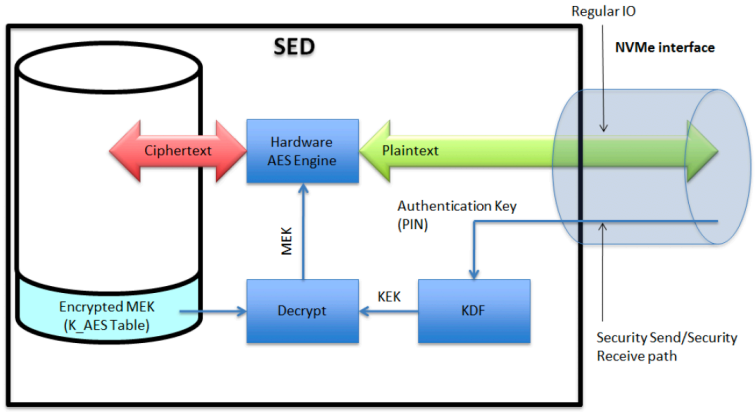
1) Background
2) KPIO SSC Overview
   - Use Cases
   - Key Provisioning
   - Security Features
   - Benefits
3) NVMe Extensions
   - Discovery
   - Key tag configuration table
   - Read/Write Operations
   - Security commands
   - Expectation for host applications
   - Error handling
4) Current state of TP-4055 proposal
5) How can you get involved?
6) Key Takeaways

© 2020 NetApp, Inc. All rights reserved.

NetApp

Background Information

- Self Encrypting Drive is a Storage Device that incorporates encryption of user data at rest:
  - Encrypts all user accessible data written to the storage device all the time, at interface speeds. The data is decrypted as it is read.
  - Media Encryption Keys are generated/held by the storage device.
  - Encryption scope is based on contiguous LBA ranges per namespace basis.
  - Passwords (AK) delivered to SED in the clear.



Copyright © 2004 – 2015 Trusted Computing Group® (www.trustedcomputinggroup.org.) All Rights Reserved
© Copyright 2007 - 2015 NVM Express, Inc. (www.nvmexpress.org.) All Rights Reserved

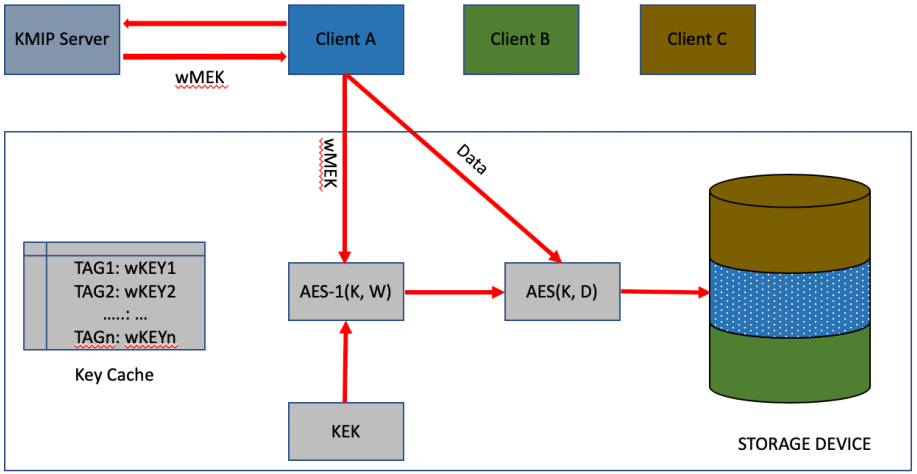
© 2020 NetApp, Inc. All rights reserved.

Disclaimer

- The KPIO SSC standardization aspects are still work in progress across NVMe and TCG Work Groups (WG)
- TP-4055 is the technical proposal for defining the NVMe interface to interact with KPIO SSC.
- TCG Storage Security Subsystem Class: Key Per IO, Version 1.00 specification is in draft stage.
  - Draft 1.01, September 5, 2019.
- The KPIO SSC is designed in such a way it could be used either with NVMe or SCSI devices.

© 2020 NetApp, Inc. All rights reserved.

Key Per IO – Concept Overview



- KEK (Key Encryption Key) and MEKs (Media Encryption Keys) are sourced from external key manager by the host
- TCG command structure are used to inject the following:
  - Initial KEK is sent to establish the secure interactions.
  - Subsequently, Key Encryption Key (KEK) is updated by inserting into the SED (unique KEK on per namespace basis).
  - MEKs are wrapped by KEK (namespace specific) and inserted into the SED by mapping it to a “Key Tag” by host software.
- Upon successful processing of TCG commands by the SED, wrapped MEKs are associated with the Key Tag/namespace ID in volatile key cache within the SED.
- Subsequent IO can use the “Key Tag” to encrypt/decrypt data to/from the storage device in a non-contiguous fashion
- Media Encryption Keys are not stored in non-volatile memory of the drive
- Crypto erase accomplished by deleting the key from the Key Manager and purging the key from key cache, which can be accomplished on per tenant basis.

© 2020 NetApp, Inc. All rights reserved.
NetApp

KPIO Security Subsystem Class (SSC)

- A Key Per IO Security Subsystem Class (SSC) compliant Storage Device (SD) will support the following capabilities:
- Facilitates feature discoverability.
- Provides some user definable features:
  - Access control
  - Key configuration
  - User passwords, etc.
- Supports Key Per IO SSC unique behaviors:
  - Communication
  - Configuration management

© 2020 NetApp, Inc. All rights reserved.

KPIO SSC Use Cases

- Deploy Storage Device & Take Ownership:
  - The Storage Device is integrated into its target system and ownership transferred by setting or changing the Storage Device's owner credential.
- Activate or Enroll Storage Device:
  - Key Encryption Keys (KEKs) and key cache slots, and authentication credentials are configured. Access control (access lock) is configured for KEK access.
- Lock & Unlock Storage Device:
  - Offers more flexibility to unlock or lock specific ranges according tenant ownership.
- Media Encryption Key injection:
  - Media Encryption Keys (MEKs) are injected and associated with specific key slots which are later specified during device IO for the purpose of encryption and decryption of relevant user data within the Storage Device.
- Repurpose & End-of-Life:
  - Erasure of KEKs and reset of KEK, purging of key cache and access credential(s) for Storage Device repurposing or decommissioning.

© 2020 NetApp, Inc. All rights reserved.

Key Provisioning

- Key provisioning comprises of following steps, executed in the following ordered sequence.
- KEK provisioning
  - Provisioning and Update of the KEK is protected by access control mechanisms similar to TCG Opal.
  - Authentication to the Admin Authority within the SED must occur first in order to provision the initial KEK in plaintext and may be subsequently updated via wrapping with the previously provisioned KEK.
  - Only a single KEK is being stored in the SED at a time, the storage device shall support a unique KEK for each namespace provided by the storage device
- MEK/DEK provisioning
  - Assumes that a KEK has previously been inserted
  - MEK/DEK may be generated and formatted (Tag, Type, Length, Value: TTLV) for object encoding by an external key management system such as OASIS KMIP Server, by wrapping it with the KEK and passed into the storage device.
- Key tagging
  - MEK/KEKs passed into the storage device may be referenced by future IO in the same controller through the use of key tags by the host software.
  - The specific key tag mechanism will leverage NVMe directives, when directive type is set to “key tag”.

© 2020 NetApp, Inc. All rights reserved.

Security Capabilities

- A storage device compliant with the Key Per IO SSC SHALL support password authorities and authentication, similar to SED.
- The securely downloaded and wrapped MEK/DEK's are maintained in volatile memory of the NVM subsystem and therefore purged when powered off providing stronger confidentiality assurance against physical theft/loss of device
- A storage device complying with Key Per IO SSC shall support Full Disk Encryption for all host accessible user data stored on media using AES-256 encryption algorithm.
- A storage device compliant with the Key Per IO SSC SHALL support at least two Security Providers (SPs):
  - Admin SP
  - Key Per IO SP
- Initial access control policies are preconfigured at storage device manufacturing time on manufacturer created SPs.
  - A Key Per IO SSC compliant storage device SHALL support personalization of certain Access Control Elements of the Locking SP

© 2020 NetApp, Inc. All rights reserved.

KPIO SSC - Benefits

- The KPIO proposal allows encryption keys to be managed and downloaded securely by the host to the NVM subsystem.
- Encryption of user data is based on each I/O command indication which key to utilize for encryption of that command.
- This provides a finer granularity of data encryption that enables a granular encryption scheme in order to support the following use cases:
  - Easier support of European Union's General Data Protection Regulations' (GDPR): "Right to be forgotten".
  - Easier support of data erasure when data is spread over many disks (e.g., RAID/Erasure Coded).
  - Easier support of data erasure of data that is mixed with other data needing to be preserved.
  - Assigning an encryption key to a single sensitive file or a host object.

© 2020 NetApp, Inc. All rights reserved.



NVMe Extensions

© 2020 NetApp, Inc. All rights reserved.

Discovery

- Discovery allows host to discover the following:
- Identify namespace provides a hint that KPIO might be supported.
  - At controller and namespace level
- Feature support/enable status
- Granularity and alignment of encrypted I/Os
  - Per namespace and format
- TCG discovery commands (Security Send / Security Receive)
  - Authenticate with the device
  - Security Receive (security protocol = 00h)
  - Discover security protocols 0x01, 0x02, others?, 0xEA

© 2020 NetApp, Inc. All rights reserved.

Key Tag Configuration Table

- Use TCG command (Security Send)
- Associate a Key Tag with an MEK
- Load MEK and Key Tag association into the Subsystem key cache

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

© 2020 NetApp, Inc. All rights reserved.

Key Tag Configuration Table

- Use TCG command (Security Send)
- Associate a Key Tag with an MEK
- Load different MEKs and Key Tag associations into the Subsystem key cache

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

© 2020 NetApp, Inc. All rights reserved.

Read, Write, Write Zeroes, and Verify Commands

Figure 372: Read – Command Dword 12	
Bits	Description0
…	
19:16	Command Extension Type (CETYPE): Specifies the Command Extension Type that applies to the command (refer to section 8.TBD).
...	

CETYPE = 0h for DSM hints

CETYPE = 1h for KPIO

Figure 373_TBD: Read – Command Dword 13 if CETYPE is non-zero	
Bits	Description
31:16	Reserved
15:00	Command Extension Value (CEV): The definition of this field is dependent on the value of the CETYPE field. Refer to Figure NEW-6 for the definition.

CEV = Key Tag value

Figure NEW-4: Write – Command Dword 13 if CETYPE is non-zero	
Bits	Description
31:16	Directive Specific (DSPEC): Specifies the first Directive Specific value associated with the Directive Type field (refer to section 9.1).
15:00	Command Extension Value (CEV):The definition of this field is dependent on the value of the CETYPE field. Refer to Figure NEW-6 for the definition.

DSPEC = for Directives (streaming)

CEV = Key Tag value

© 2020 NetApp, Inc. All rights reserved.

Read and Write Examples

- WRITE (LBA=100, LEN=8, CETYPE=1, CEV=1)

MEK = 0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF

- WRITE (LBA=200, LEN=16, CETYPE=1, CEV=100)

MEK = 0x1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF1234567890ABCDE6

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

© 2020 NetApp, Inc. All rights reserved.

Security Send / Security Receive Commands

- Authentication
- Discovery
- Define Key Tag to MEK associations (load Key Cache)
- Define encryption / decryption algorithm
- Securely Purge Key Cache

© 2020 NetApp, Inc. All rights reserved.

Expectation for host application

- Host application has the responsibility of correctness of key/key tag injected via NVMe IO submission queues.
- Host application should prevent usage of an incorrect key in the event that a key is changed by selecting a new key tag in the list of available tags (based on information it gathered as a result of discovery) for insertion of a new key.
- If host application sends an IO request that specify an out of range (or) non-populated key tag, the drive will have to return an error.
- It is also assumed that the host application will block an IO request that requires leveraging a previously injected MEK until the MEK is successfully injected (or updated) into the drive.

© 2020 NetApp, Inc. All rights reserved.

Error Handling

- In an IO command, if a key corresponding to the key tag has not been correctly injected into the storage device using the TCG CMD, or if a key has been deleted from the storage device:
- Work in progress: The storage device shall abort the command with a status of Invalid Field in Command and media/DI error.
- This helps the drive to detect if a wrong key tag is sent with the read command when the writes were earlier committed with a different one.

© 2020 NetApp, Inc. All rights reserved.

Current State of Technical Proposal TP-4055

- Completed technical work in the committee
- Planned to begin 30-day member review soon
- Not to be publicly available until all the work (including TCG) is complete

© 2020 NetApp, Inc. All rights reserved.

What about SCSI and/or SATA

- Since the major work is in TCG …
- Yes, it would be relatively easy to add to SCSI or SATA
  - SCSI and SATA use the same TCG protocol as NVMe
- BUT – it would require completely new I/O commands
  - Such as 32 byte CDBs for SCSI (to carry the Key Tag value)
- No interest has been shown in undertaking such an effort

© 2020 NetApp, Inc. All rights reserved.

How can you get involved?

- Talk to us after the session
- Talk to your company NVMe Representative
- Talk about it in the Linux Community

- And remember, it’s still a work in progress

© 2020 NetApp, Inc. All rights reserved.

Key Takeaways

- The KPIO SSC implementation profile is defined in such a way that any SD that claims Opal SSC compatibility could conform to the KPIO SSC.
- The KPIO SSC is intended to protect the confidentiality of stored user data against unauthorized access once it leaves the owner's control (involving a power cycle and subsequent de-authentication) and enables interoperability between SD vendors.
- By virtue of its fine-grained approach towards securing specific regions of the media relevant to a tenant/client, it is possible to support multi-tenancy requirements for storage devices utilizing SED technology.

© 2020 NetApp, Inc. All rights reserved.



© 2020 NetApp, Inc. All rights reserved.