SSD Implementation of Key Per IO

Dan Helmick
Principal Architect

SAMSUNG

Overview of Key Per IO (KPIO)

Host Submits a Command

SQE provides Key Tag

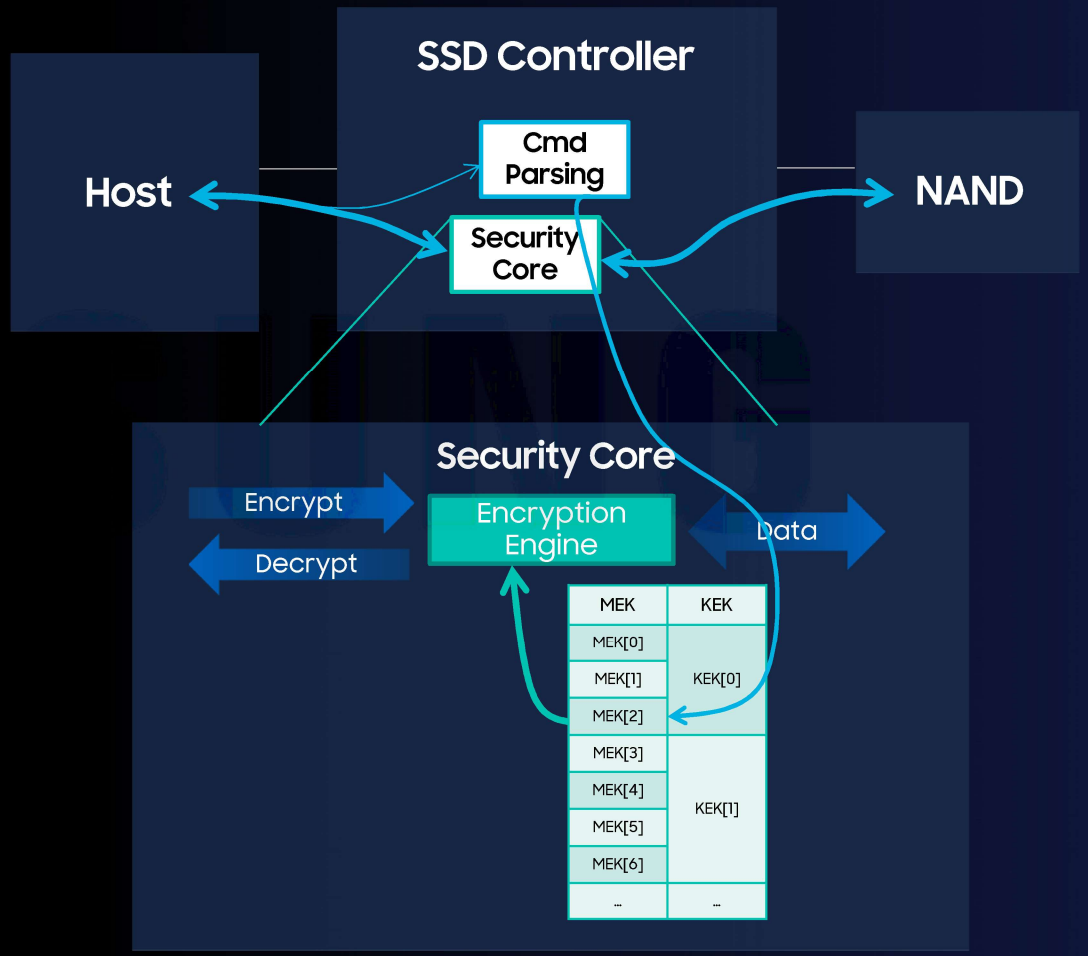
- Potentially different Key Tag for every IO

Drive

- Maintains separated security boundary
- Confirms KEK
- Encrypts/Decrypts Data with MEK

Acronyms

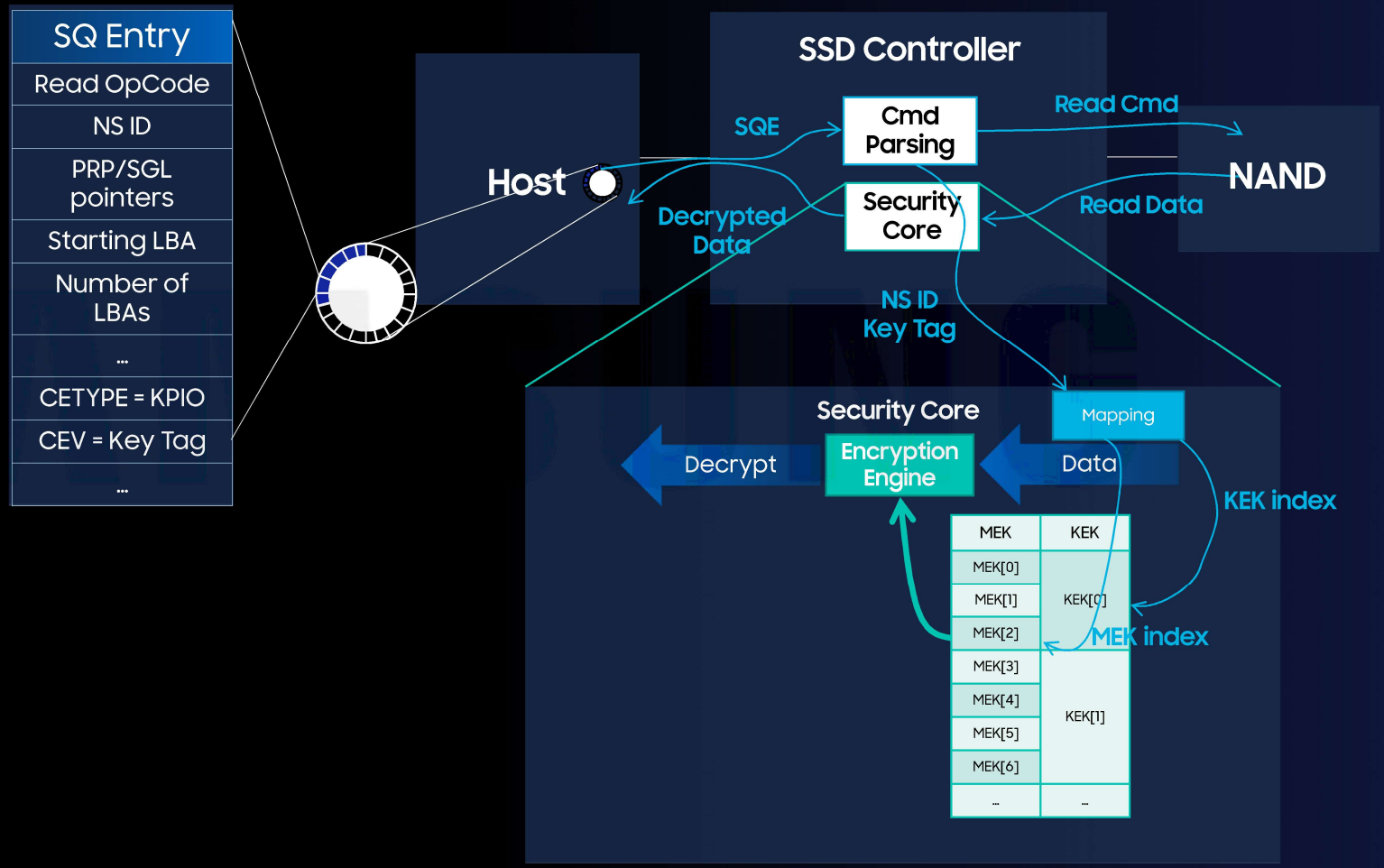
- KEK = Key Encryption Key
- KPIO = Key Per IO
- MEK = Media Encryption Key
- NS = Namespace
- SQE = Submission Queue Entry
- UID = Unique ID



SAMSUNG

KPIO Nominal Read Flow Detail

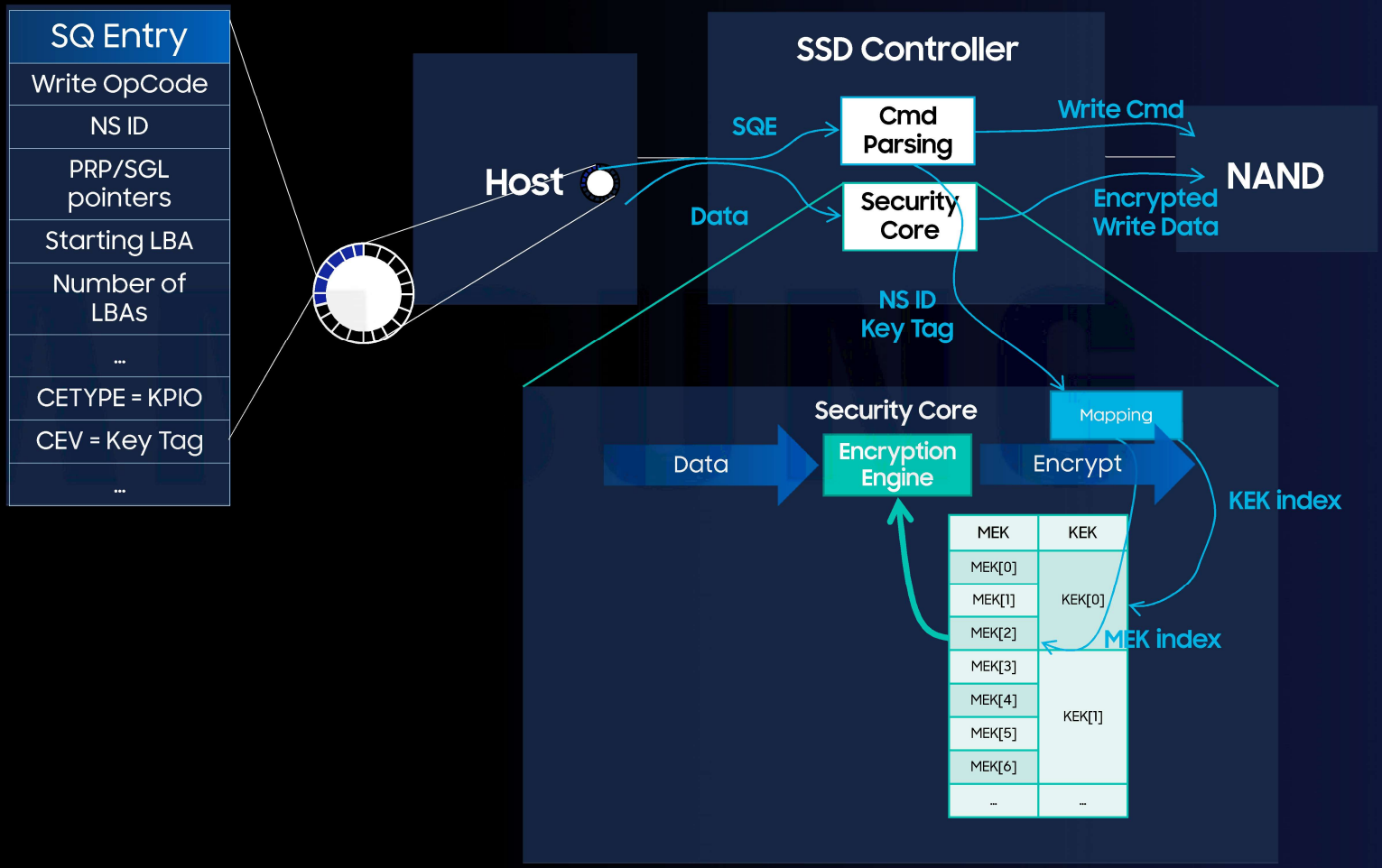
1. Host submits SQE to drive
2. SSD Parses SQE
3. Read Command proceeds and eventually returns the Read Data
4. NS ID and Key Tag are passed to the Security Core
5. NS ID and Key Tag are mapped to KEK and MEK by the Security Core
6. KEK status is confirmed
7. Read Data arrives at the Encryption Engine, and Security Core provides correct MEK
8. Decrypted data is sent to the Host



SAMSUNG

KPIO Nominal Write Flow Detail

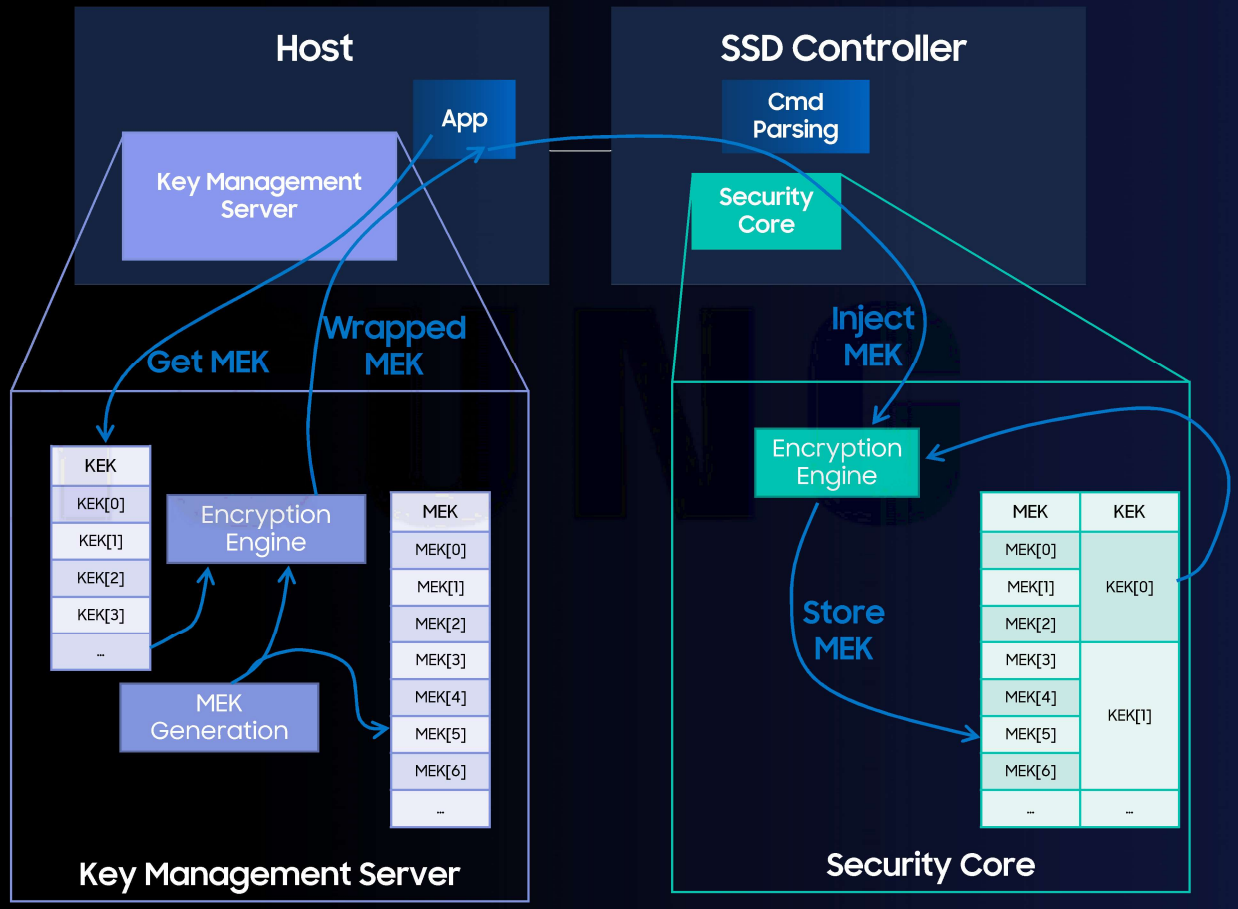
1. Host submits SQE to drive
2. SSD Parses SQE
3. NS ID and Key Tag are passed to the Security Core
4. NS ID and Key Tag are mapped to KEK and MEK by the Security Core
5. KEK status is confirmed
6. Write Command proceeds to initiate data transfer to the SSD
7. Write Data arrives at the Encryption Engine, and Security Core provides correct MEK
8. Encrypted data is stored in the NAND



SAMSUNG

MEK Injection

1. Get MEK
   - Input: KEK_UID
2. Key Management Server uses KEK to encrypt a MEK
   - Output: Wrapped MEK
3. App injects MEK to the SSD
   - Input: KEK_UID, Wrapped MEK, Key Tag
4. SSD Uses KEK to decrypt the wrapped MEK
   - SSD's MEK and KEK table is potentially very sparse



SAMSUNG

Dan's Expectations of Common Direct Attached Configurations

1st generation drives will be used to enable infrastructure and familiarize end customers

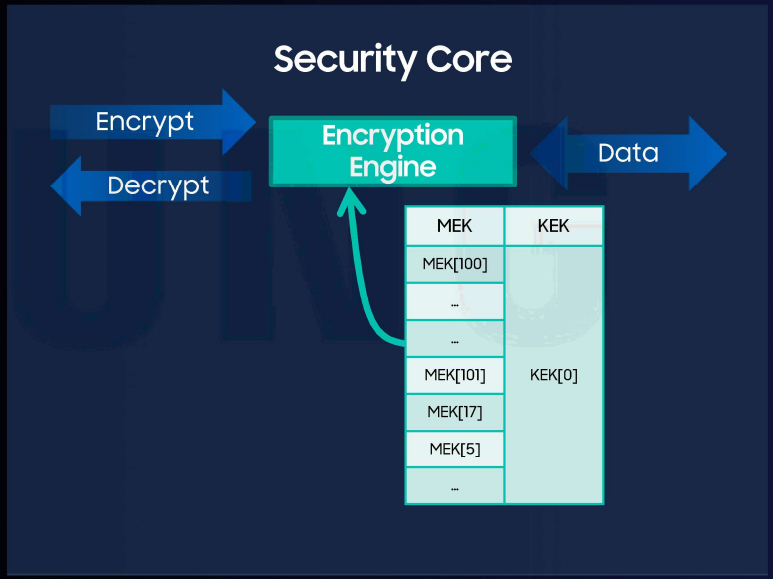
- Nuanced and exotic uses can be integrated into this base implementation

1 KEK per Drive

- Multiple KEKs seems most useful for NVMe-oF

MEK Slots == TCG OPAL key slots

- Today's use-cases for TCG OPAL use up to 1024 keys
- KPIO enables a dynamic swapping of keys extending the utility of the existing slots



SAMSUNG

Dan's Predicted Future Extensions of Direct Attached SSD Use-Cases

Supporting 2 or more KEKs per SSD

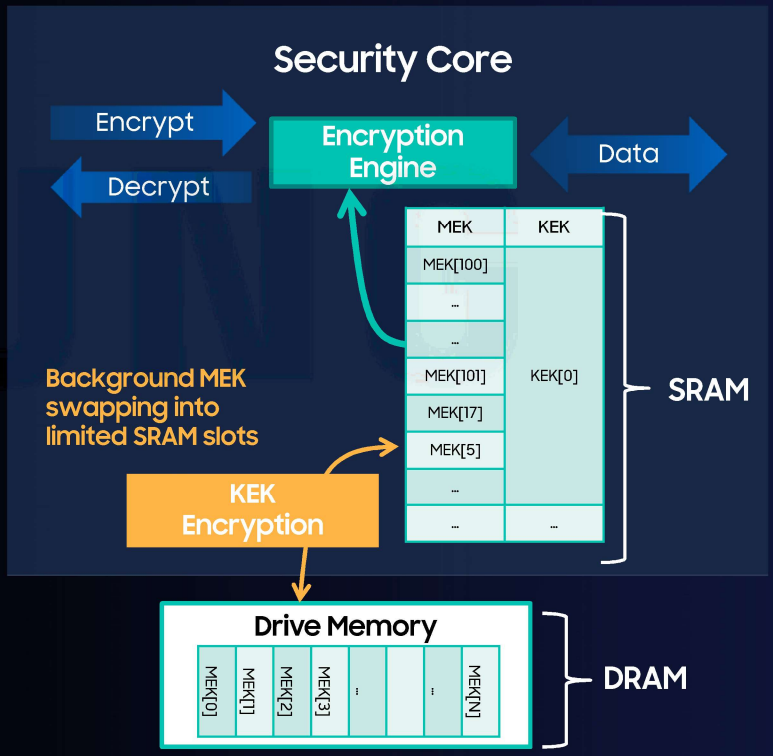
- Enables swapping KEKs
  - Enables drive reuse
  - Enables Sustainability

Using KEK to encrypt MEKs in an extended DRAM

- No Host or spec changes required for enablement
- Maintains low latency for those MEKs in SRAM
- Maintains security sub-drive boundary while enabling general DRAM sharing with unsecured SSD components
- Increases latency variation while swapping MEKs from DRAM into SRAM
  - Latency Variations are commonly unacceptable in Enterprise SSDs

Permanently associating Key Tag to NVMe attributes

- Associations enable secure legacy tenants without requiring population in every SQE
- Example associations: NS or NVM Controller
- In a Multi-Tenant environment, enable the association to be controlled by:
  - Parent - Legacy enablement
  - Child - Data ownership



SAMSUNG

Extending KPIO to Secure LM Data Transfer

Existing Live Migration (LM)

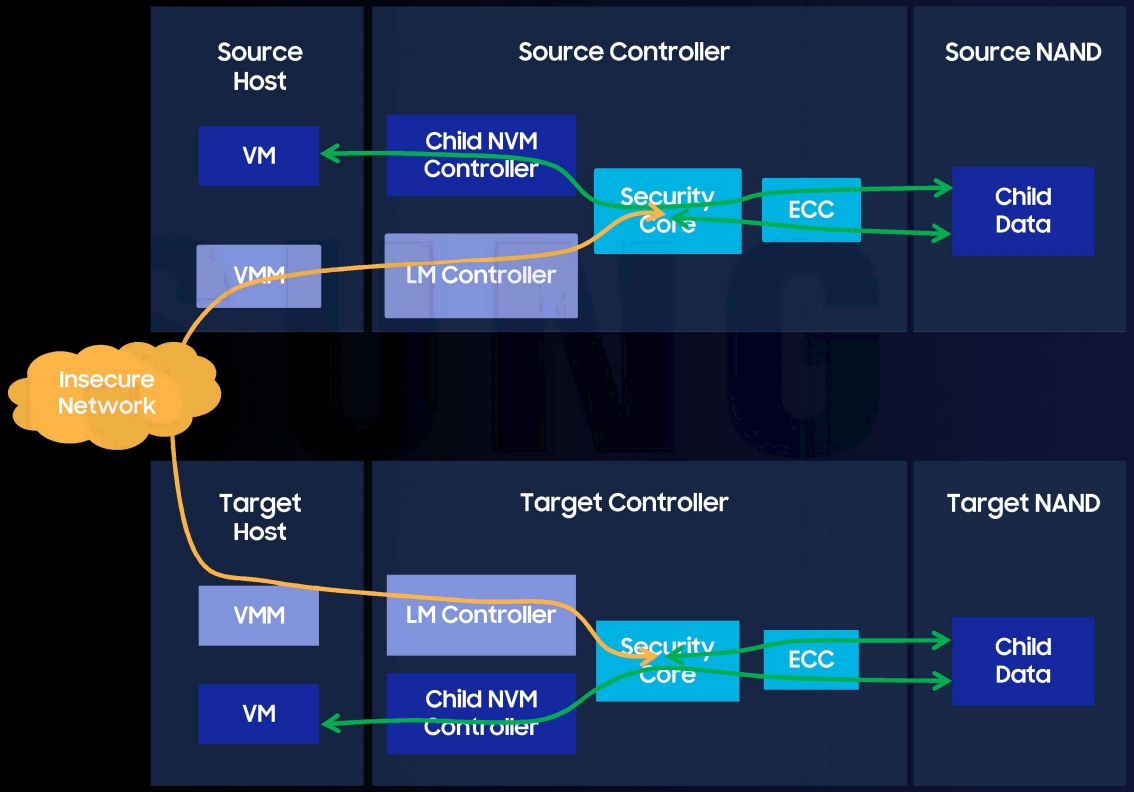
- Virtual Machines (VMs) are secured
  - Processors: Trusted Execution Environments (TEE)
  - PCIe: Integrity and Data Encryption (IDE)
  - Storage: OPAL and KPIO
- VM Manager (VMM) and network data transfers are unquantified risk

Extending KPIO into LM

- Allows different ECCs, CRCs, and MetaDatas on Source and Target SSDs
- Assures Encryption Engine interoperation with MEKs
- Key Management can be owned by VM
- Legacy VMs could be enabled by a Key association
- Prevents VMMs or Networks from viewing data

Items to solve during standardization

- Preferred Encryption Tweak during the VMM data movement
- ???



SAMSUNG

Conclusions

KPIO is an extensible feature set enabling better end customer security

1st generation KPIO SSD enablement

- Centered around 1 KEK
- Uses KPIO to dynamically swap MEKs and extend utility of existing TCG OPAL key slots

Future KPIO Extensibility Ideas

- Assignment of Key for legacy tenants
- Live Migration of VM data while maintaining encryption
- Encrypted Snapshots

Thank You