AKM CYBER CORP

AUTOMATING CYBERSECURITY

# AKM Cyber Corp.

## Scalable Cyber Security from Endpoint to Enterprise

*Best, True, “Zero Trust Architecture” (ZTA)*

| | | |
|---|---|---|
| CEO: Bart Shields, | bart@akmcyber.com, | +1 (951) 522 3540 (USA) |
| COO-Engr: Bill Basser | bill@akmcyber.com, | +1 (931) 302 7397 (USA) |
| Advisor: Bruce McIndoe | bruce@mcindoeriskadvisory.com, | +1 (410) 320-4513 (USA) |

7000 Columbia Gateway, Suite 150 Columbia, MD 21046

# AKM Cyber Has the Solution

AKM Cyber's core technology: **Autonomous Key Management** (**AKM**) is a breakthrough technology that simplifies and automates the entire cybersecurity landscape that will enable companies to secure their networks, endpoints, and assets in ways never before thought possible.

AKM Cyber has two broad offerings:

1) **Autonomous Key Management** (**AKM**) – Designed to be a drop-in replacement for PKI, AKM autonomously manages networks of devices such that once a network cybersecurity framework has been put in place, it never needs to be configured again and can run for perpetuity without ever needing operator assistance while still refreshing all security credentials on a periodic basis.
2) **Asset Integrity Management** (**AIM**) – This is an AKM extension that implicitly provides a Zero Trust Architecture without any of the ongoing complexity required by ZTA today, and ensures all assets (both hardware and software) within a network are continuously verified.

AKM and AIM are autonomously managed and can run continuously without operator intervention. Both dramatically simplify the cybersecurity configuration to nothing more than network configuration. Both utilize true "Security by Design" principles that ensure the confidentiality and integrity of your entire network.

# What is Autonomous Key Management (AKM)?

- **AKM is a ‘Zero Knowledge’ digital asset-centric cryptographic cybersecurity framework** that prevents most cyber risks from occurring in the first place.

- **AKM eliminates the need for many of today’s remediation-oriented ‘Band-Aid’ solutions**.

- **AKM operates autonomously** and can be applied to concurrent virtualized groupings of users, devices and digital assets. Each grouping is referred to as, an AKM Trust Relationship (ATR).

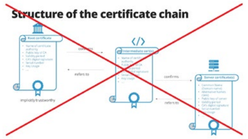
- **AKM is highly resilient to quantum computers and human error**, and far more flexible, administratively efficient, and cost-effective than todays standard PKI + TLS protocols.

**Autonomous Key Management (AKM) = Crypto Key Management System (KMS) + Secure Communication Protocol**

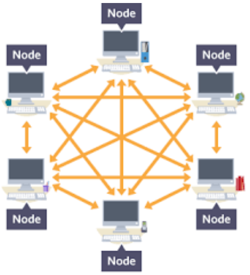
# What is Our Value Proposition?

## AKM – Autonomous Key Management

- No External Certificates

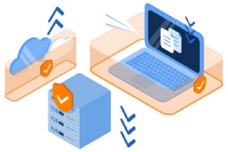


- True End-to-End Security for both Point-to-Point and Multipoint-to-Multipoint Networks



- Encrypted Data-in-Motion and Data-at-Rest

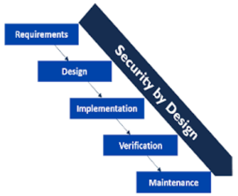
- Authentication and Integrity Monitoring



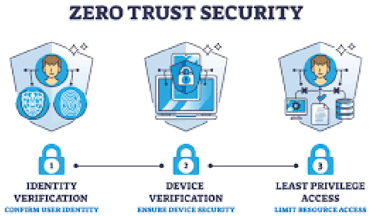
- Quantum Resilience – No Secrets are ever exposed

- Less Complexity, Lower Costs, Easier to Manage, More Secure

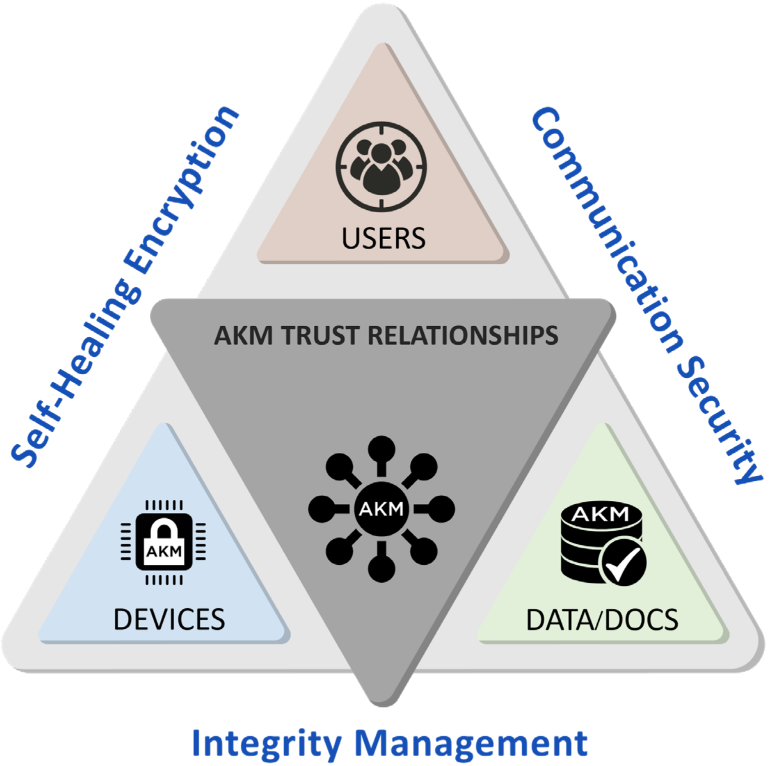
- True Security by Design



- Simplified Zero Trust Architecture



# Understanding AKM a Little Deeper



1. Turns any physical (connected) or virtual (digital) asset into an **AKM Protected Asset (APA)**, issuing each a unique AKM-ID

2. Configures APAs into **any number* of AKM Trust Relationships (ATRs)** of any type, scale, or structure, with each representing a unique AKM Trust Relationship

3. Each ATR concurrently operates its own **pre-configured, contextualized security rules** while being 100% cryptographically isolated from every other ATR

4. Once configured, each independent **ATR autonomously manages itself**.

5. In addition to configuration, the **AKM Management Module** can monitor and collect heuristic data and potentially **self-heal an ATR** in case of an attempted breach.

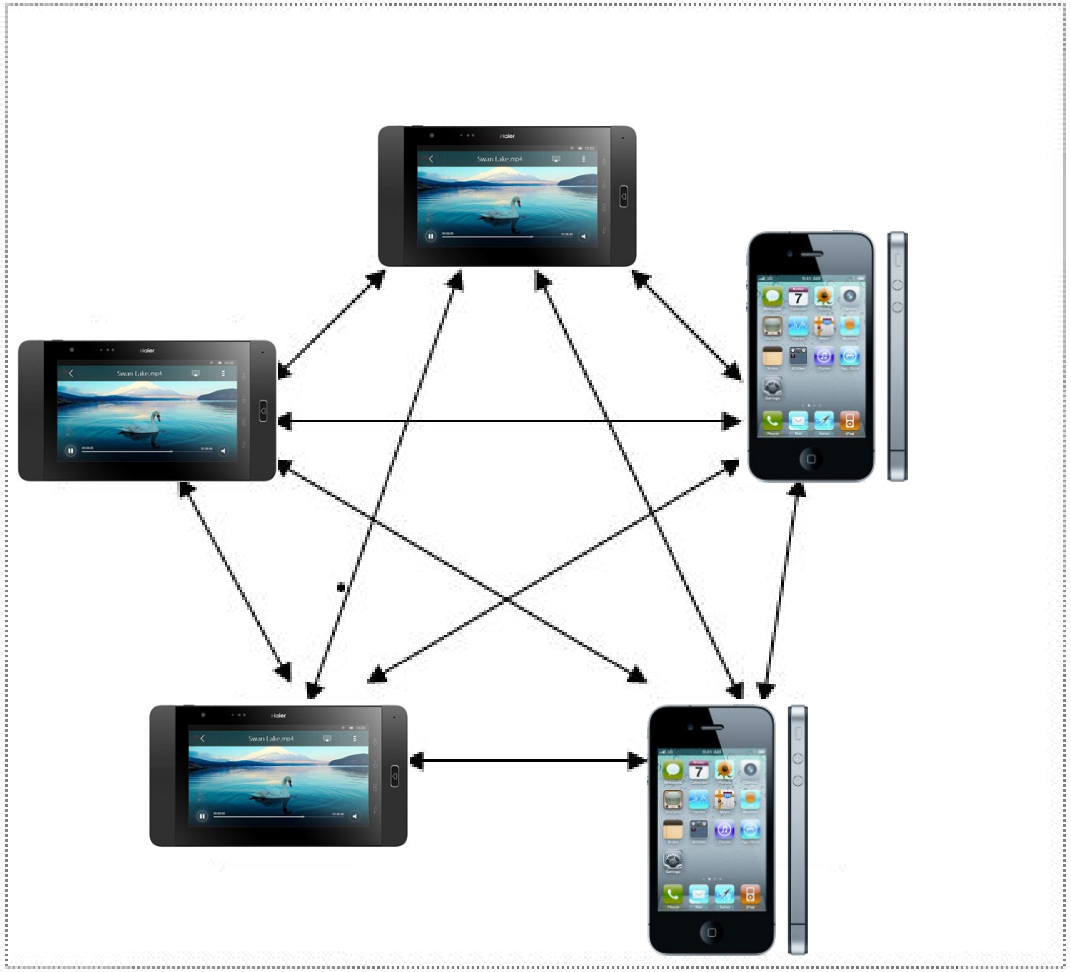
6. An extension of the AKM Endpoint SDK, the **AKM Asset Integrity Management (AIM) (i.e., an asset-monitoring service)** can also be enabled.

** Up to the implementation-specific limit.*

# AKM End-to-End, Multipoint-to-Multipoint Communication

- All  ATRs Share a Common Set of Security Credentials, referred to as a Synchronized Data Set (SDS).
- ATRs may be Air Gapped from external connections without losing any security capabilities, given that they all update their own copies of the SDS independently.
- ATRs may be Continuously Monitored through the use of an AKM Management Module (local to the network)
- Automatic Rules-based Failure Recovery that are individually tailored to each ATR
- Each ATR provides a unique Micro-Segmentation Capability per ATR
- AKM Endpoints may support multiple ATRs
- ATRs provide Implicit Zero Trust, with significantly less complexity and overhead as compared to how it is done in PKI-based systems.
- AKM ATRs provide a Simplified Network Security Model

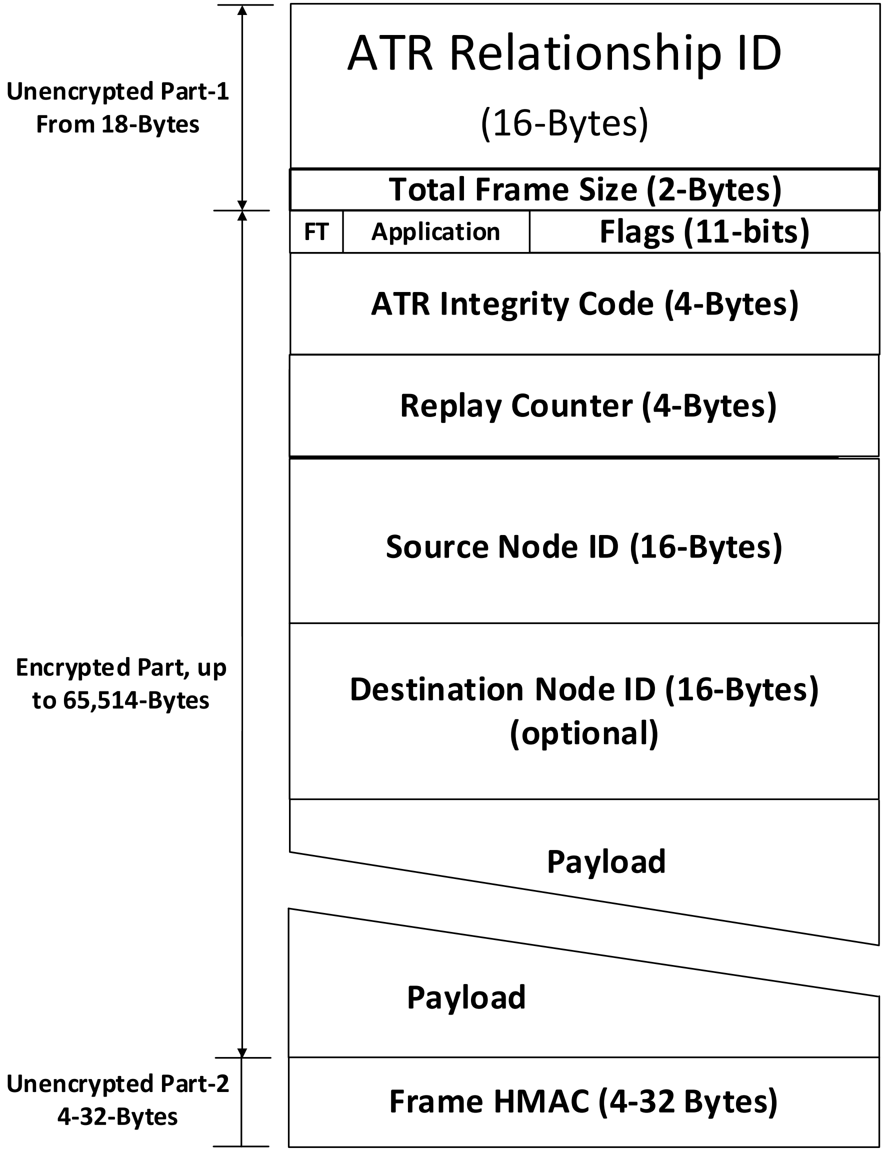
## AKM Trust Relationship (ATR)



Peer to peer multi-point communication

# AKM Data Frame

- **ATR Relationship Identifier** – AKM Identifier is specific to the ATR and is guaranteed to be unique from any other AKM Identifier.
- **Total Frame Size** – Number of bytes of entire frame, starting with the ATR Relationship Identifier and ending with the Frame HMAC.
- **Frame Type (FT) Bit** – Bit indicates whether the frame is a data or management frame.
- **Application Bits** – User-defined bits identifying the specific user application to which the frame belongs. Used to delineate applications between hosts.
- **Flags** – TBD bits defining different variations of the frame structure.
- **ATR Integrity Code** – Used to provide a zero-trust framework between endpoints.
- **Replay Counter** – Provides a counter to protect against replay attacks.
- **Source Node Identifier** – AKM Identifier, identifying the source of the frame.
- **Optional Destination Node Identifier** – AKM Identifier, identifying the recipient of the frame. Not needed in PTP configurations, but in multipoint-to-multipoint, may be used to delineate between broadcast and directed frames.
- **Payload** – Application Data
- **Frame HMAC** – used to authenticate the frame's integrity and source.



Unencrypted Part-1 From 18-Bytes

ATR Relationship ID (16-Bytes)

Total Frame Size (2-Bytes)

FT | Application | Flags (11-bits)

Encrypted Part, up to 65,514-Bytes

ATR Integrity Code (4-Bytes)

Replay Counter (4-Bytes)

Source Node ID (16-Bytes)

Destination Node ID (16-Bytes) (optional)

Payload

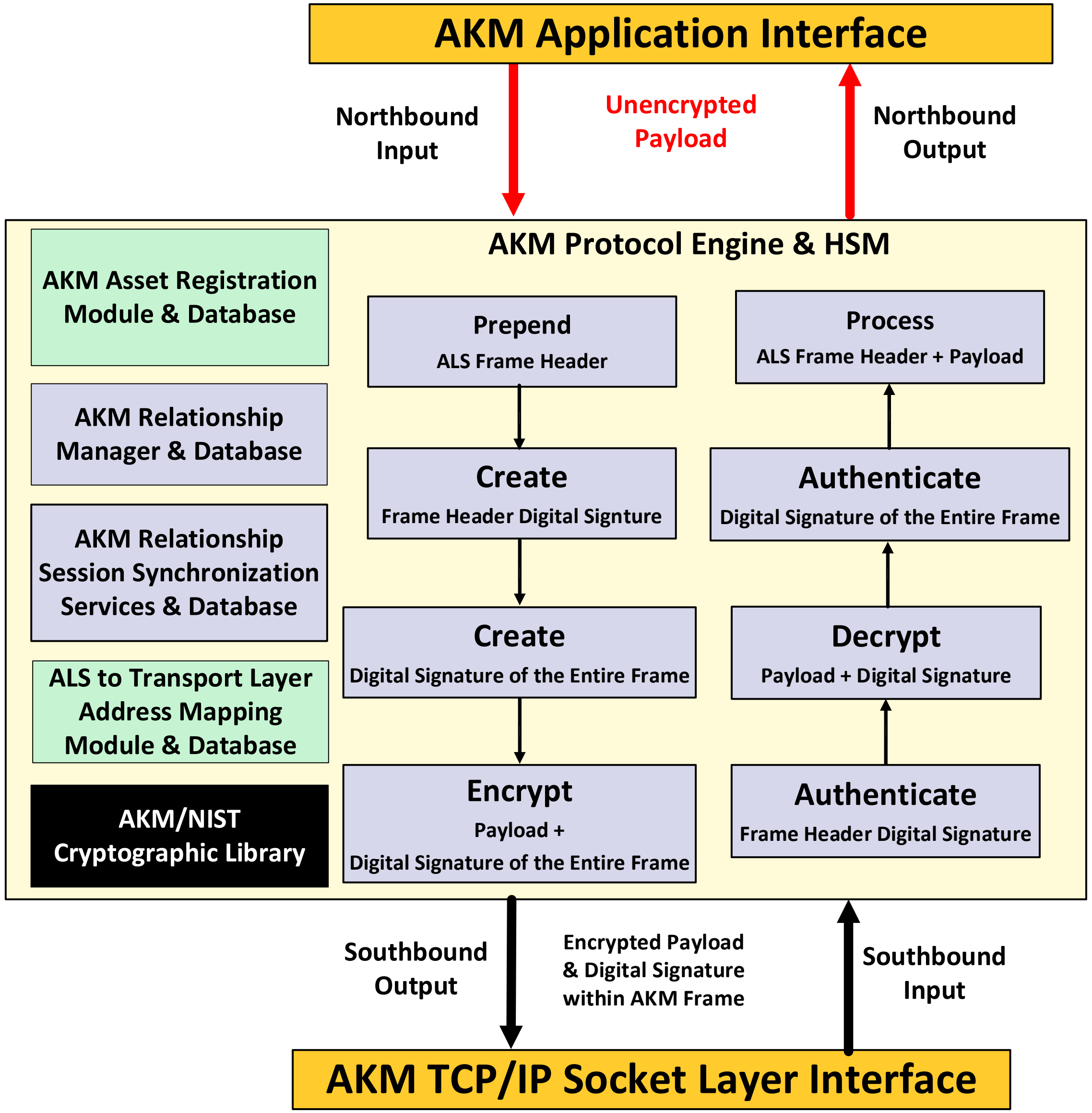
Payload

Unencrypted Part-2 4-32-Bytes

Frame HMAC (4-32 Bytes)

# AKM Crypto-Accelerator HSM

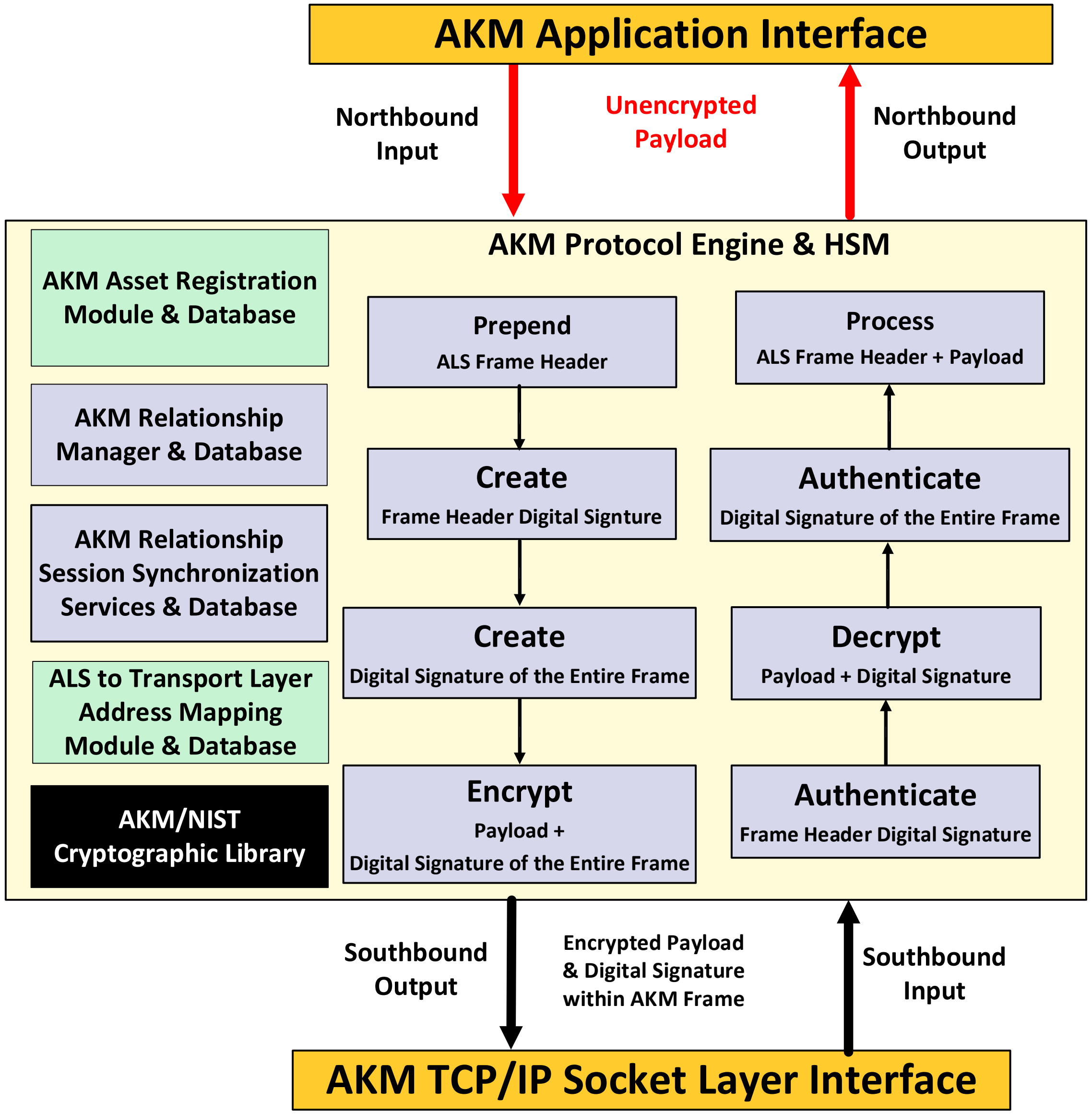
- The AKM Crypto-Accelerator HSM has a red port and a black port. The red port communicates unencrypted data and related application information between itself and the applications above. The black port communicates AKM Data Frames and related information between the AKM Layer Security (ALS) Protocol and the AKM TCP/IP Socket Layer below.
- A key point of this architecture is that the module is physically separate from the host system and never shares any AKM security credential information with the host. All encryption and decryption (and authentication-related processing) is done internally within the AKM Crypto-Accelerator HSM. Thus, even if malware somehow infects the host system, it cannot infect the AKM Crypto-Accelerator HSM.
- Also, because AKM updates security credentials internally, no security credential information from the AKM Crypto-Accelerator HSM is ever shared externally. Thus, no opportunity for a breach and quantum resilience by design.
- Proposed HW Design of AKM Protocol Engine & HSM is to be implemented within an FPGA, to take advantage of any sequential algorithms (such as the AKM Finite State Machine (FSM) and NIST library functions) that can offloaded to HW programming. Additional implementation of FPGA soft core for management functions done in ‘C’, and within an lightweight RTOS (ex. Free RTOS).


AKM Application Interface
Northbound Input
Unencrypted Payload
Northbound Output
AKM Protocol Engine & HSM
AKM Asset Registration Module & Database
AKM Relationship Manager & Database
AKM Relationship Session Synchronization Services & Database
ALS to Transport Layer Address Mapping Module & Database
AKM/NIST Cryptographic Library
Prepend ALS Frame Header
Create Frame Header Digital Signture
Create Digital Signature of the Entire Frame
Encrypt Payload + Digital Signature of the Entire Frame
Process ALS Frame Header + Payload
Authenticate Digital Signature of the Entire Frame
Decrypt Payload + Digital Signature
Authenticate Frame Header Digital Signature
Southbound Output
Encrypted Payload & Digital Signature within AKM Frame
Southbound Input
AKM TCP/IP Socket Layer Interface

# AKM Crypto-Accelerator HSM

**The SDS (shown in the below diagram) and its contents are never shared outside of the HSM. Thus, guaranteeing all security credentials remain 100% secure between endpoints.**

| |
|---|
| <Security Relationship ID$_{\text{ATR-A}}$> |
| <AKM ATR Integrity Code$_{\text{ATR-A}}$> |
| <AKM Current Session Credentials$_{\text{ATR-A}}$> |
| <AKM Next Session Credentials$_{\text{ATR-A}}$> |
| <AKM Fallback Session Credentials$_{\text{ATR-A}}$> |
| <AKM Fallback Next Session Credentails$_{\text{ATR-A}}$> |
| <AKM Failsafe Session Credentials$_{\text{ATR-A}}$> |
| <AKM Failsafe Next Session Credentials$_{\text{ATR-A}}$> |
| <ATR List of Endpoints in Ascending Order$_{\text{ATR-A}}$> |



AKM Application Interface

Northbound Input

Unencrypted Payload

Northbound Output

AKM Protocol Engine & HSM

AKM Asset Registration Module & Database

AKM Relationship Manager & Database

AKM Relationship Session Synchronization Services & Database

ALS to Transport Layer Address Mapping Module & Database

AKM/NIST Cryptographic Library

Prepend
ALS Frame Header

Create
Frame Header Digital Signture

Create
Digital Signature of the Entire Frame

Encrypt
Payload +
Digital Signature of the Entire Frame

Process
ALS Frame Header + Payload

Authenticate
Digital Signature of the Entire Frame

Decrypt
Payload + Digital Signature

Authenticate
Frame Header Digital Signature

Southbound Output

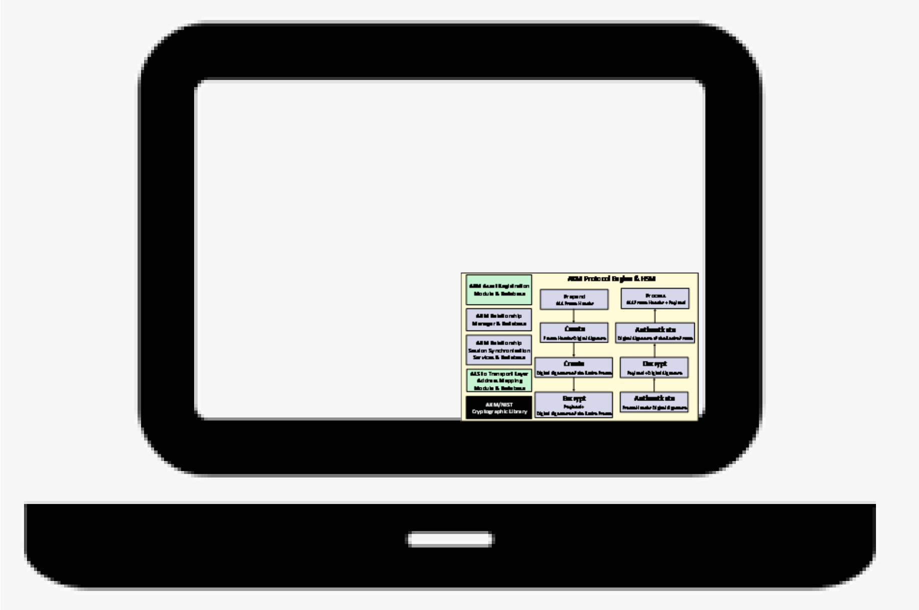
Encrypted Payload
& Digital Signature
within AKM Frame

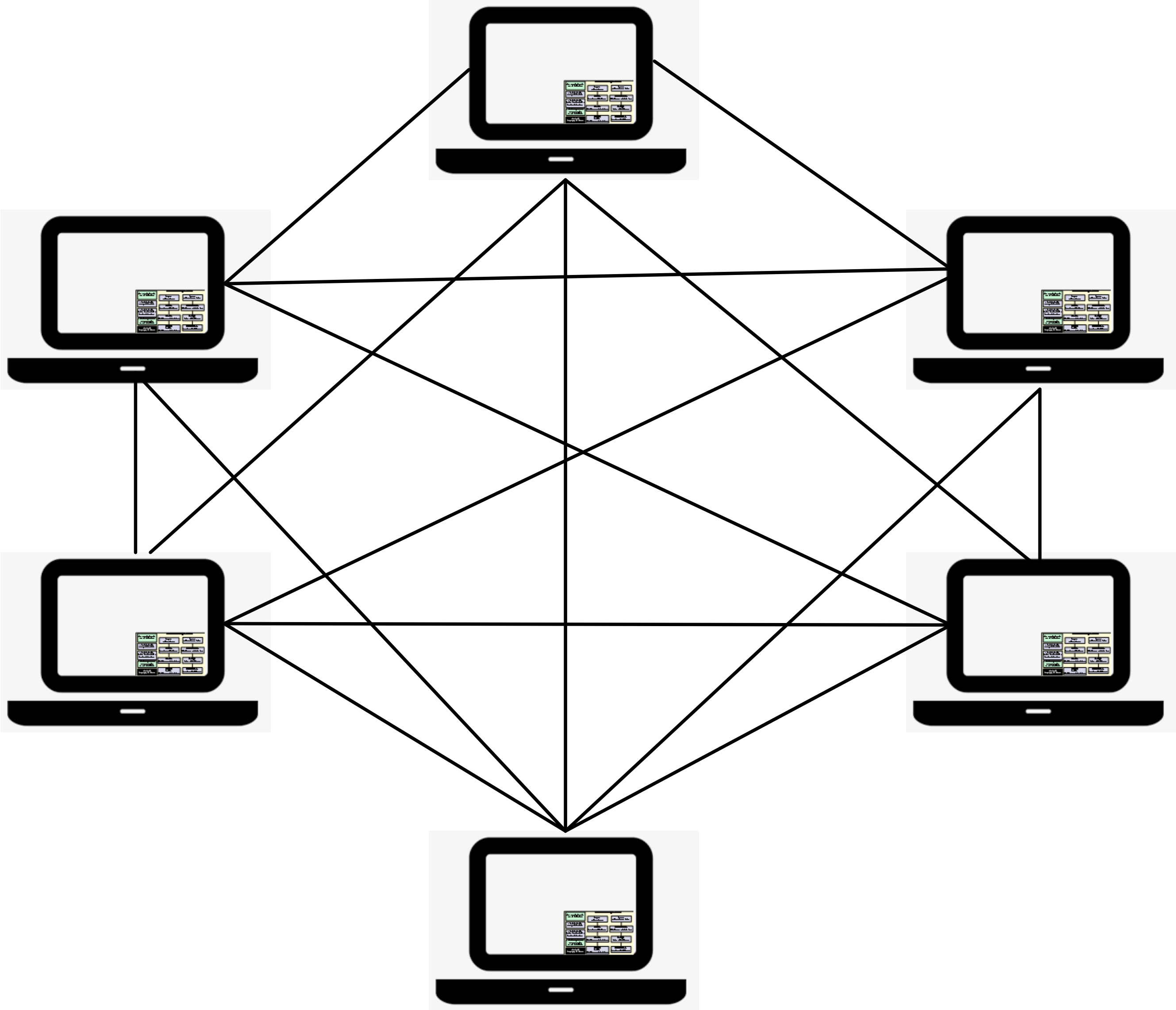
Southbound Input

AKM TCP/IP Socket Layer Interface

# AKM Crypto-Accelerator HSM

## Example AKM Network using an AKM Crypto-Accelerator HSM within each AKM Endpoint.

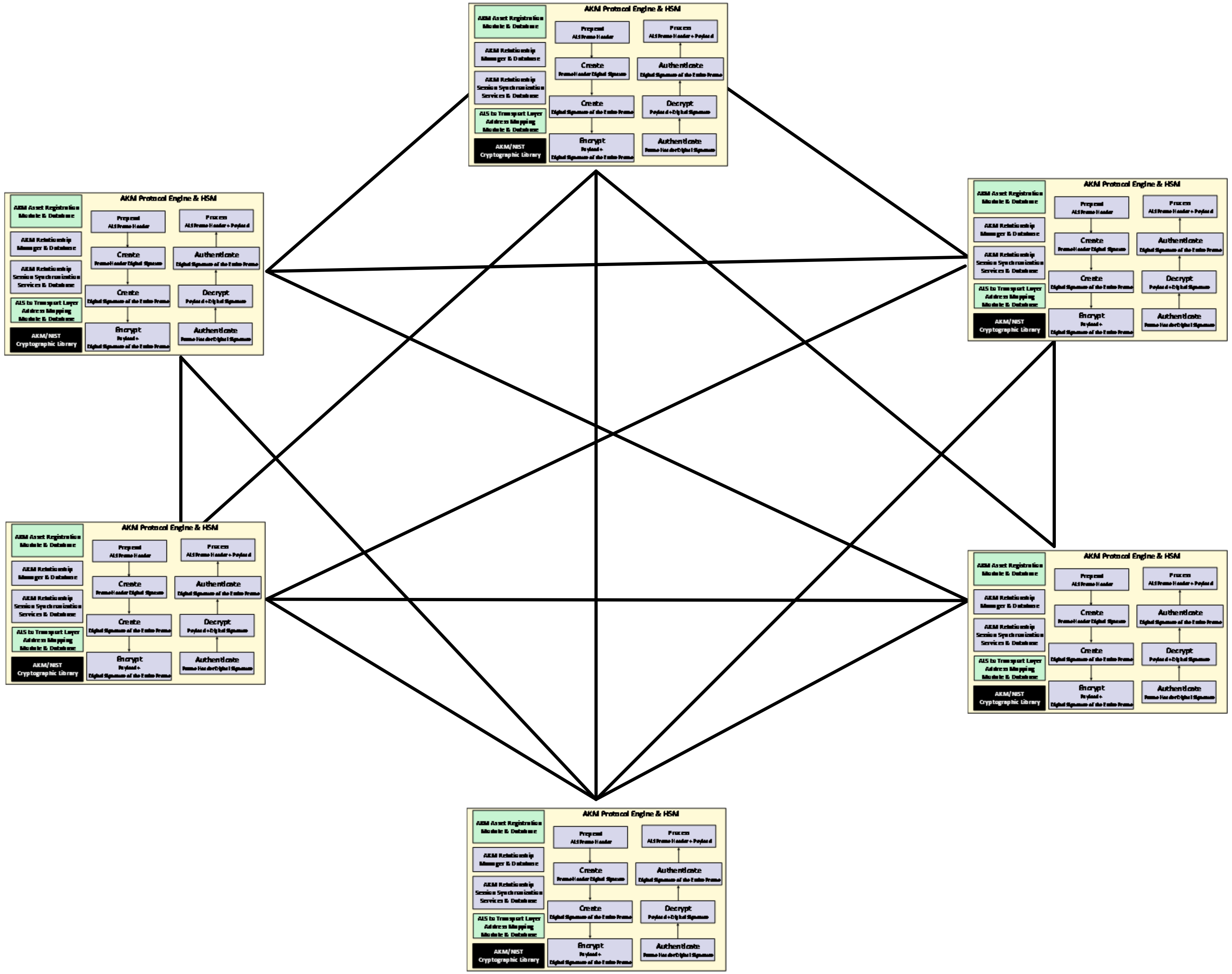

AKM Protocol Engine & HSM
AKM Asset Registration Module & Database
AKM Relationship Manager & Database
AKM Relationship Session Synchronization Services & Database
AKS to Transport Layer Address Mapping Module & Database
AKM/NIST Cryptographic Library
Prepend AL Frame Header
Process AL Frame Header + Payload
Create Frame Header Digital Signature
Authenticate Digital Signature of the Entire Frame
Create Digital Signature of the Entire Frame
Decrypt Payload + Digital Signature
Encrypt Payload + Digital Signature of the Entire Frame
Authenticate Frame Header Digital Signature



# AKM Crypto-Accelerator HSM

Because all encryption and decryption operations are done internally within each laptop's onboard AKM Crypto-Accelerator HSM, communication between laptops having an AKM Crypto-Accelerator HSM within them is essentially, HSM to HSM direct communication. Thus, implying there is absolutely zero chance of a breach.

No secrets come in or out. What happens within each AKM HSM remains within each AKM HSM. Making AKM truly quantum resilient.


AKM Protocol Engine & HSM
AKM Asset Registration Module & Database
AKM Relationship Manager & Database
AKM Relationship Session Synchronization Services & Database
ALS to Transport Layer Address Mapping Module & Database
AKM/NIST Cryptographic Library
Prepend ALS Frame Header
Create Frame Header Digital Signature
Create Digital Signature of the Entire Frame
Encrypt Payload + Digital Signature of the Entire Frame
Process ALS Frame Header + Payload
Authenticate Digital Signature of the Entire Frame
Decrypt Payload + Digital Signature
Authenticate Frame Header Digital Signature

# Form Factor

Actual Form Factor is TBD, but more than likely target M.2 form factor, with a PCIe interface.  Thus, making it laptop friendly.