Enterprise Key Management Infrastructure Technical Committee (EKMI TC)

IEEE 1619.3 Briefing

Arshad Noor
Chair, EKMI TC
arshad.noor@strongauth.com

The Encryption Problem

- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy

- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy

- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy

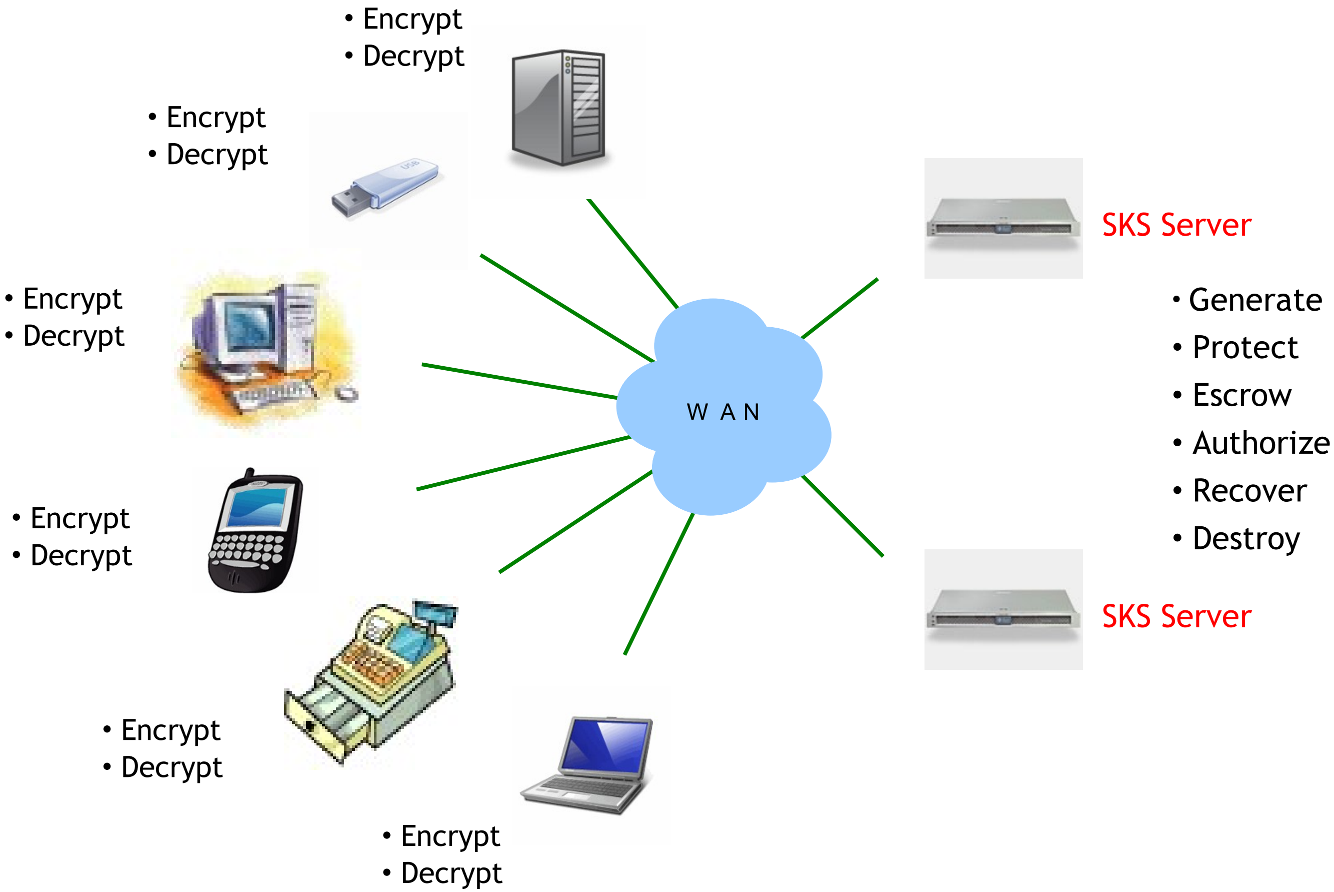
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy

- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy

- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy

.........and on and on

The Encryption Solution



- Encrypt
- Decrypt

- Encrypt
- Decrypt

- Encrypt
- Decrypt

- Encrypt
- Decrypt

- Encrypt
- Decrypt

- Encrypt
- Decrypt

WAN

SKS Server

- Generate
- Protect
- Escrow
- Authorize
- Recover
- Destroy

SKS Server

What is an EKMI?

- An **Enterprise Key Management Infrastructure** is:

  "A collection of technology, policies and procedures for managing ***all*** cryptographic keys in the enterprise."

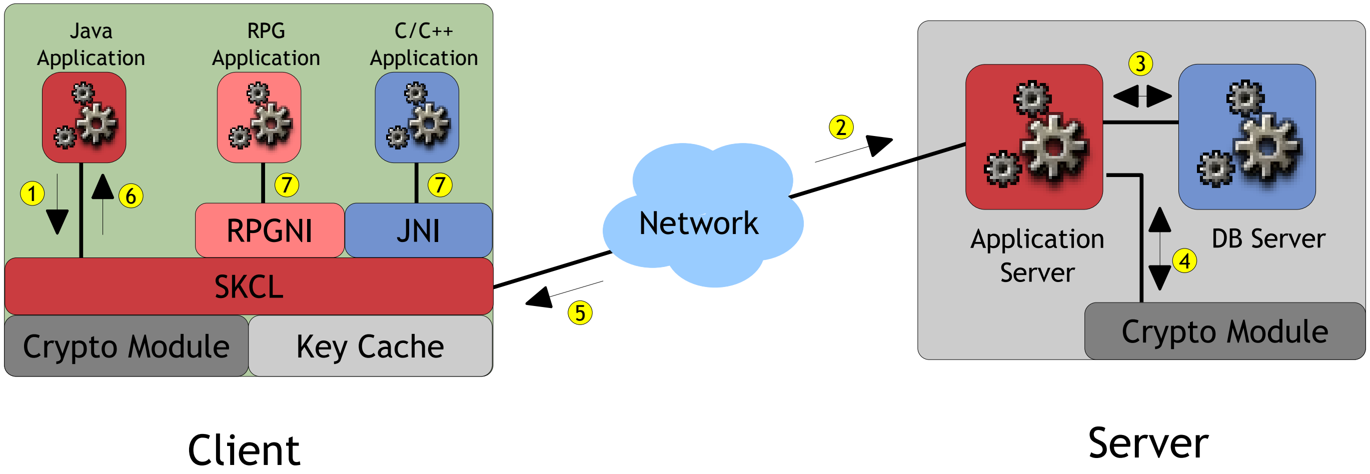
EKMI Characteristics

- A single place to define EKM policy
- A single place to manage all keys
- Standard protocols for EKM services
- Platform and Application-independent
- Scalable to service millions of clients
- Available even when network fails
- Extremely secure

EKMI Components

- Public Key Infrastructure
  - For digital certificate management; used for strong-authentication, and secure storage & transport of symmetric encryption keys
- Symmetric Key Management System
  - **SKS Server** for symmetric key management
  - **SKCL** for client interactions with SKS Server
- **EKMI = PKI + SKMS**

SKMS Big-Picture



1. Client Application makes a request for a symmetric key
2. SKCL makes a digitally signed request to the SKS
3. SKS verifies SKCL request, generates, encrypts, digitally signs & escrows key in DB
4. Crypto HSM provides security for RSA Signing & Encryption keys of SKS
5. SKS responds to SKCL with signed and encrypted symmetric key
6. SKCL verifies response, decrypts key and hands it to the Client Application
7. Native (non-Java) applications make requests through Java Native Interface

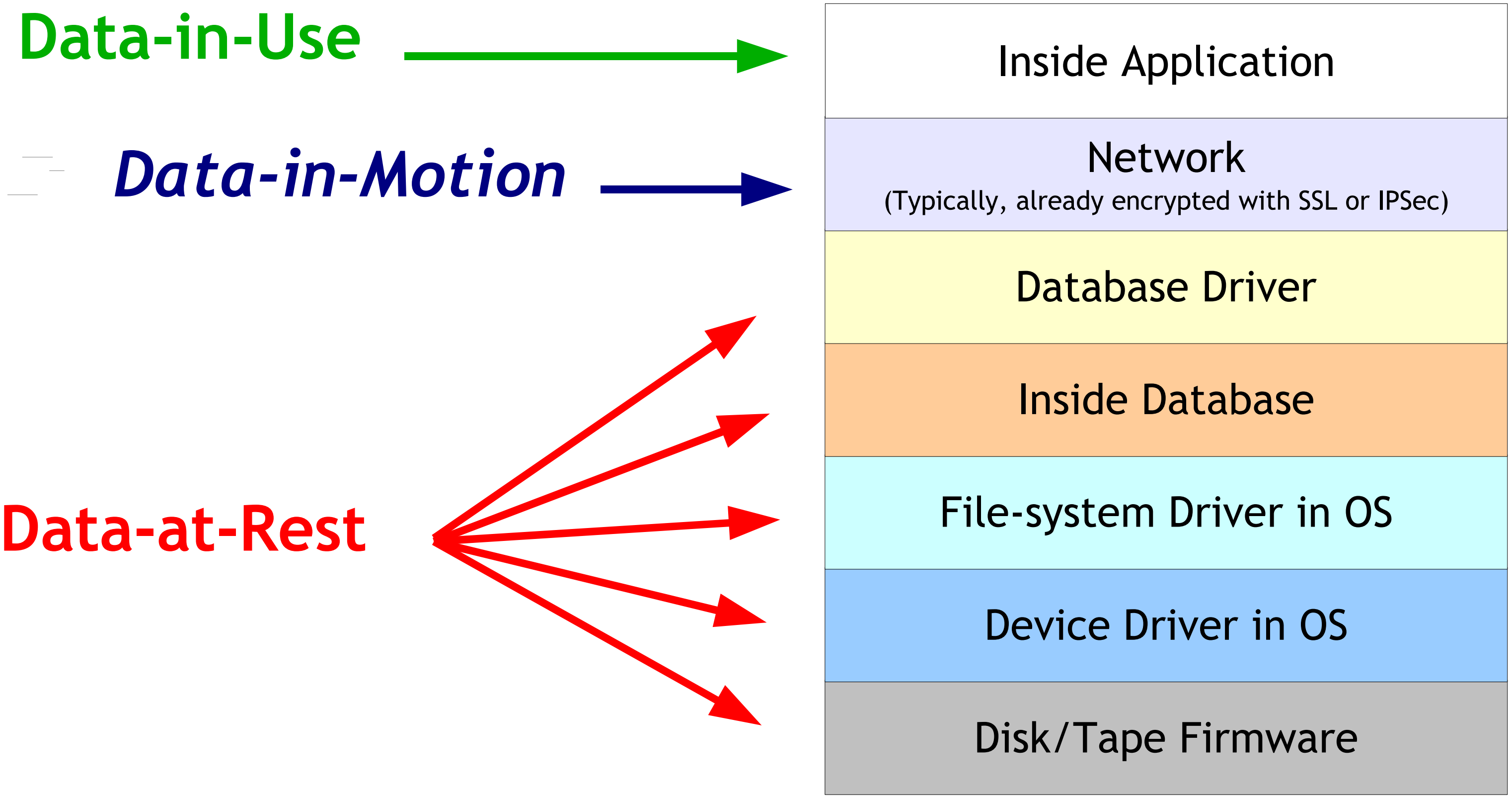
EKMI TC Goals

- Standardize on a Symmetric Key Services Markup Language (SKSML)
- Create Implementation & Operations Guidelines
- Create Audit Guidelines
- Create Interoperability Test-Suite

EKMI TC Members/Observers

- FundServ, PA Consulting, PrimeKey, Red Hat, StrongAuth, US DoD, Visa, Wave Systems
- Many large companies as Observers
  - Security, Database, Consulting, Non-US Government Agency
- Individuals representing Audit and Security backgrounds

Potential Encryption Layers



Data-in-Use

Data-in-Motion

Data-at-Rest

Inside Application

Network

(Typically, already encrypted with SSL or IPSec)

Database Driver

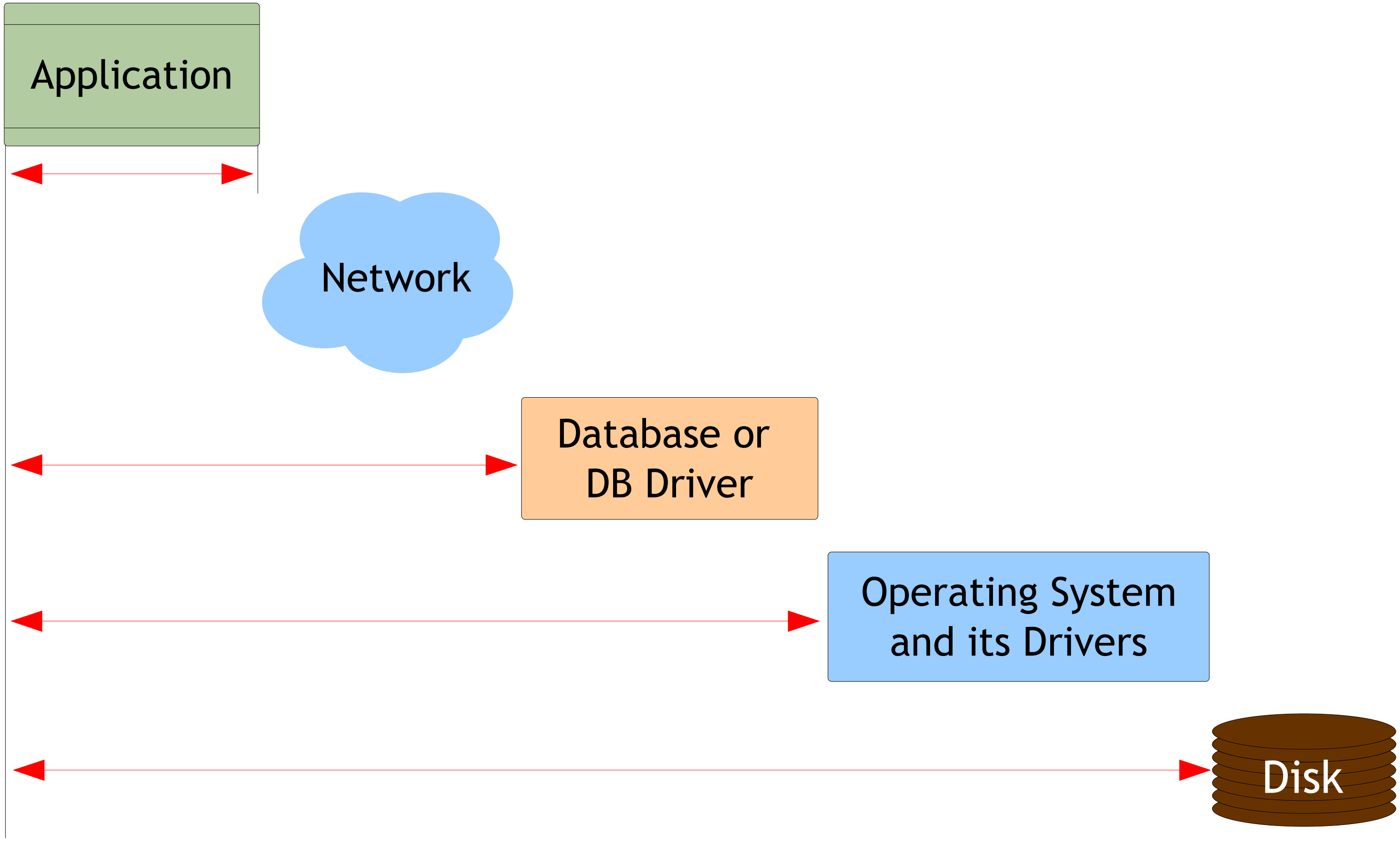
Inside Database

File-system Driver in OS

Device Driver in OS

Disk/Tape Firmware

Exposure Spread



Application

Network

Database or DB Driver

Operating System and its Drivers

Disk

Vulnerability due to exposure of unencrypted data

Potential IEEE & OASIS integration?

- Incorporate SKSML into management consoles (MC) that control devices
  - MC becomes an SKSML Client
  - Use SKSML to acquire keys and policies from SKS Server
  - Use IEEE standards/protocols for pushing keys from MC to devices
- Other mechanisms?

Resources

- OASIS EKMI TC Resources
  - Use Cases, SKSML Schema, Presentations, White Papers, Guidelines, etc.
- www.strongkey.org - Open Source SKMS implementation
- www.issa.org - Article on SKMS in February 2007 issue of ISSA Journal