Geoff Rousell, Senior IT Specialist, zChampion, geoff_rousell@uk.ibm.com

05 November 2009

Key Management & Disk Encryption

Session DJ – GSE Conference 2009

Synopsis

- “So they said you had to encrypt the data at rest.
  - Did you ever think to ask 'Why?'
  - Did you ever think what would be the 'best' way?
  - Did you think someone else would solve it for you?
  - Did you ever think that you would be the one to manage it?”

- If you answered “YES”, “NO” or “I DON'T KNOW” to any of these questions this session should help you to think about these issues and decide whether disk encryption is right for you.
- This is not a technical session about cryptography, but describes practical data confidentiality and how you might manage it.

Agenda

- Disk Encryption
  - Why should I do it?
  - What is it?
  - How can I do it?
- Key Management
  - What is it?
  - Why do I need it?
  - How might I implement it?
- Summary, best practices

Assumptions

- You've probably attended session DF – z/OS cryptography, and session DA – Securing FTP and Telnet
- You know the difference between symmetric and asymmetric encryption
- You've come with questions to have answered, or at least discussed!

Disk Encryption

Disk* Encryption – Why should I do it?

- Increasing demands for data confidentiality
  - Legislative
  - Compliance
- Increasing amounts of data held online
  - Decreasing cost
  - Increasing online data mining of operational data
- Well established methods for keeping data confidential in transit
- Mixture of methods for ensuring confidentiality **at rest**

* Where “disk” could be something other than “brown spinning stuff”

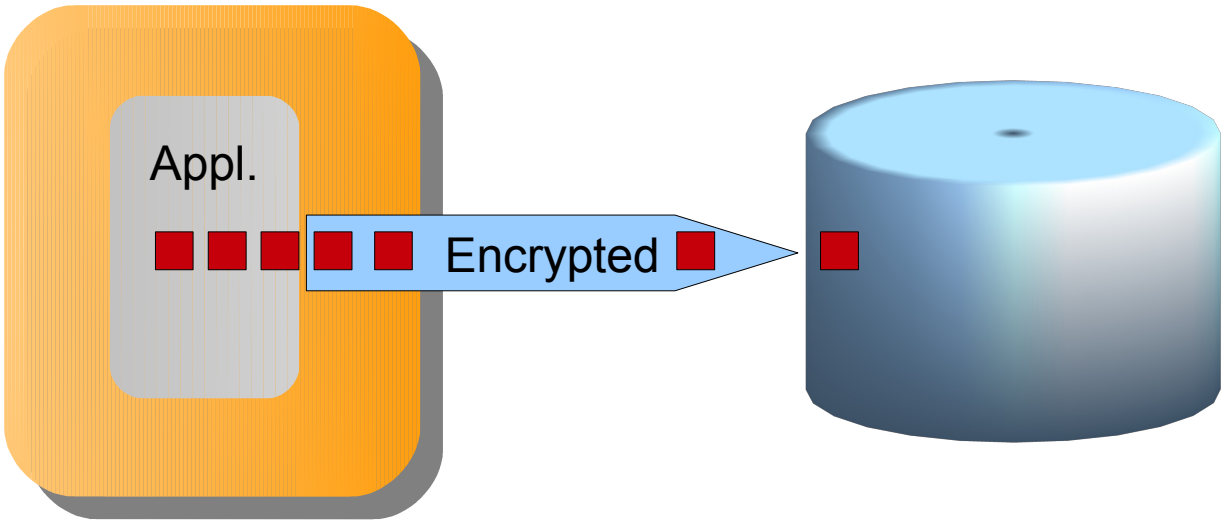
Encrypting Data at Rest

- Hardware or Software or combination
- File level or full disk (FDE)
  - Often characterised as *Application Managed* vs *System Managed*
- Lots of different implementations
  - Pure Hardware
    - Storage Hardware
      - Encrypting tape drive/library
      - Encrypting disk/controller
  - Pure Software
    - e.g. PGP
  - Hybrid, which may involve:
    - accelerators
    - co-processors
    - HSMs*

* Hardware Security Module

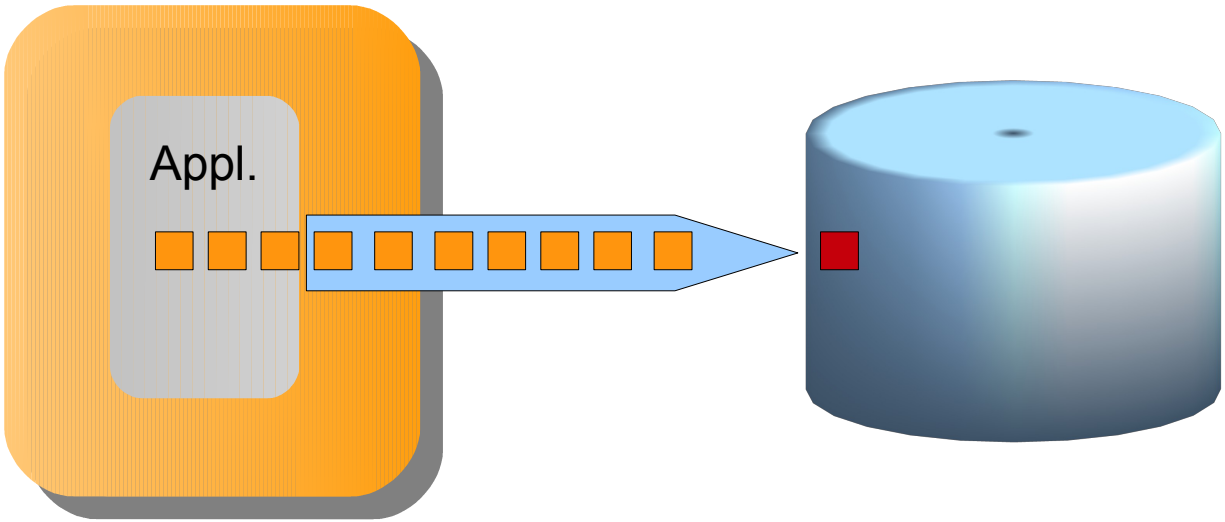
File Level or “Application Managed” encryption

- Requires knowledge of data to be encrypted
- Data classification is key
- Many different mechanisms
  - May actually be implemented in middleware
- Data encrypted from CPU all the way to storage media
- Potential performance penalty
  - Use of hardware acceleration may assist



Full Disk Encryption

- Quite fashionable
- Data encrypted as it reaches media surface
- Performance impact negligible
  - Designed to match data rate to media
- Some advantages over Application Managed Encryption
  - Transparent to applications
  - Potentially less complex
  - Reduced need for data classification



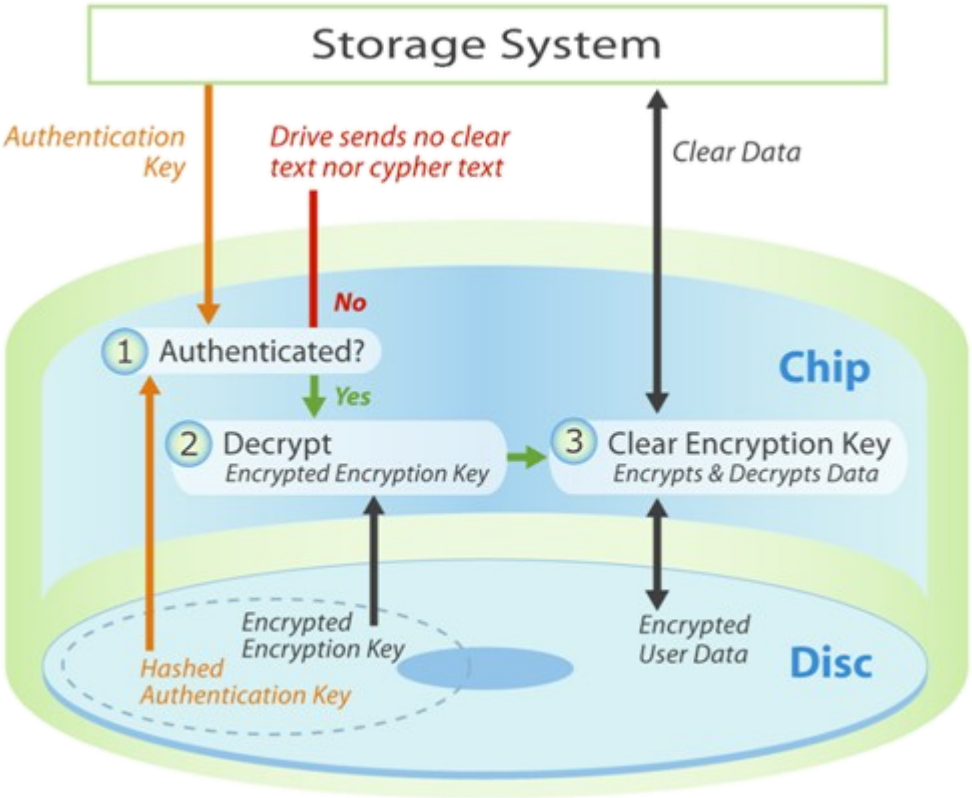
Full Disk Encryption – Why should I do it?

- What attacks are we defending against?
  - Removal of media
    - Theft
    - Maintenance – **90%** of drives we accept have readable data on them
    - End of Lease
    - End of Life

- Any other things?
  - Safe erasure
    - other approaches are time consuming, costly, error-prone or simply dangerous.




Full Disk Encryption - How



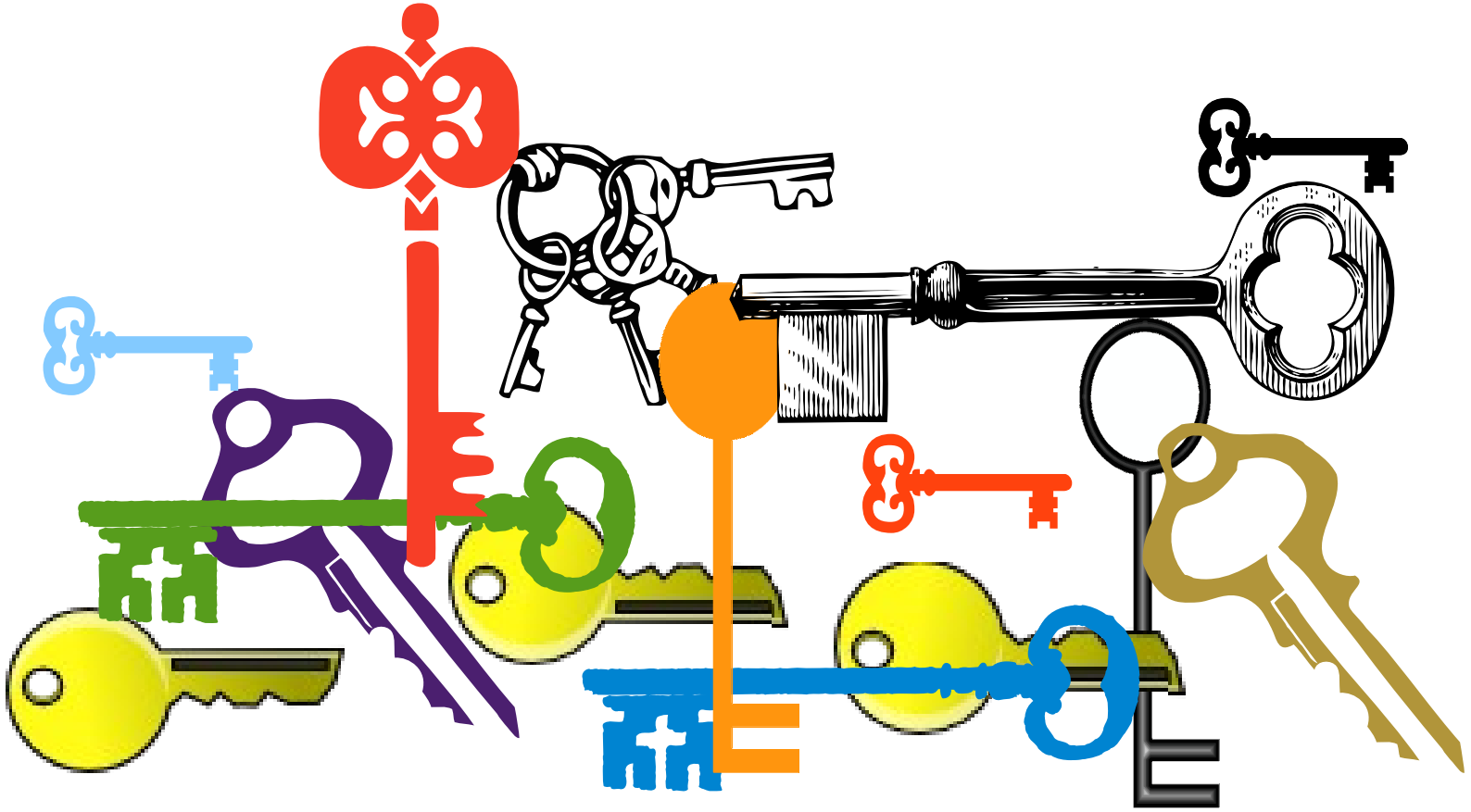
One or two things we need to worry about.

- What is the risk?
- Encryption of system startup data
  - Try make sure you can start the system
- Trusted Platform Module
  - Is the encrypting function verifiable?
  - Typically assessed against standard such as FIPS-140
- Backup & Mirroring
  - Will need similar capability at restore site

Summary of data encryption modes

	Hardware	*Subsystem*	*Application*
Protects all data at rest.	Potentially	Probably not	Probably not
Protects data backups	If backups stored on protected hardware	Probably	Yes
Protects data in System z memory	No	Largely not	Yes
Provides RACF controls on data access	No	No	Potentially
Provides regulatory compliance	Maybe	Maybe	Maybe
Can provide “Secure Key” protection	Yes	Not at present	Yes
Transparent to application code	Yes	Yes	No
Transparent to subsystem configuration	Yes	No	No
Data is “Searchable”	Yes	Yes	Possibly. Needs special design
Works with hardware data compression	Yes	Probably not	Probably not
Performance	Very Good	Good	Application design dependent

Key Management



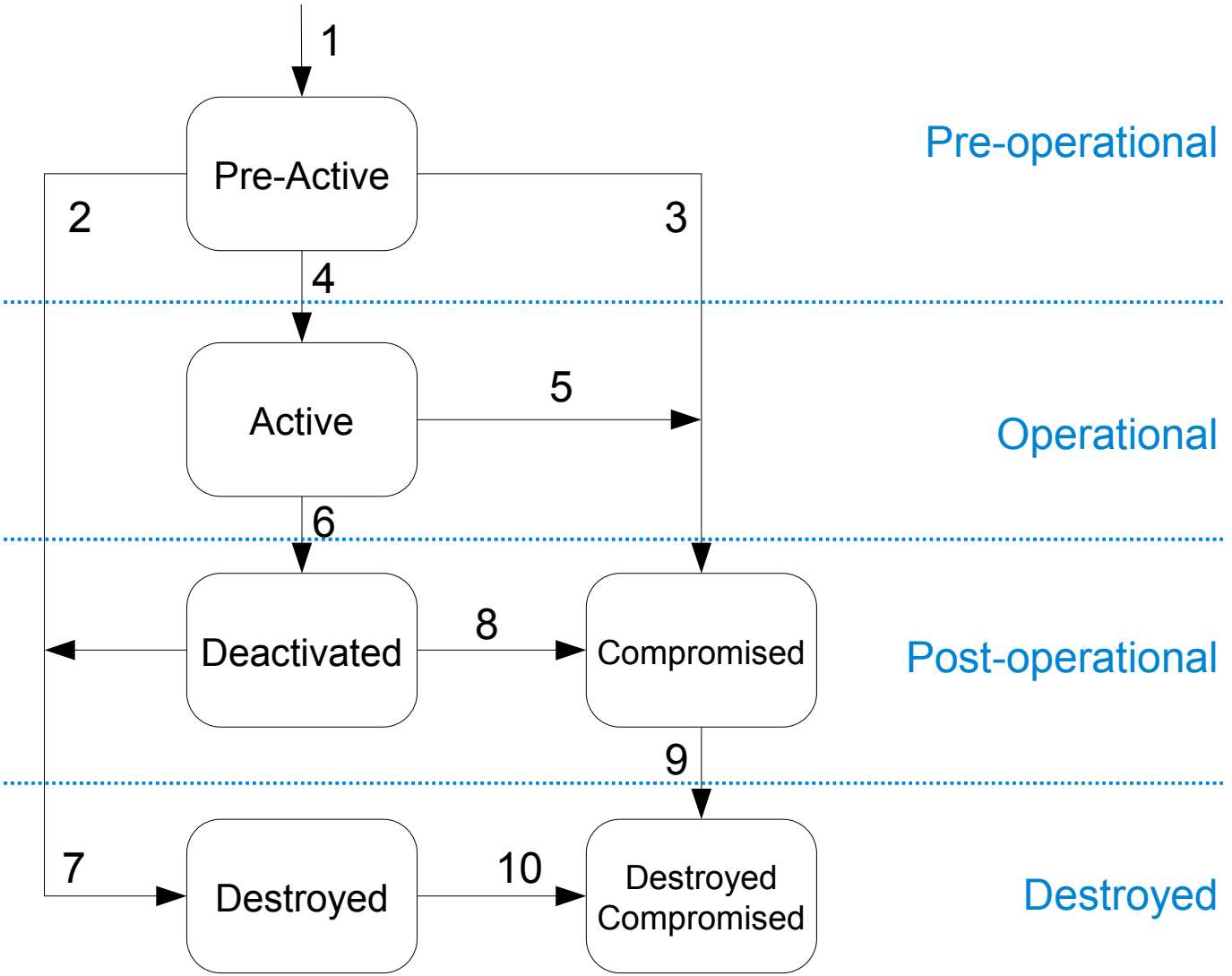
Key Management – What is it?

- According to NIST (SP800-57 Part 1, 2nd Edition):

  "The proper management of cryptographic keys is essential to the effective use of cryptography for security."

- Key Management Lifecycle
  - 4.1 User Registration
  - 4.2 System and User Initialization
  - 4.3 Keying Material Installation
  - 4.4 Key Establishment
  - 4.5 Key Registration
  - 4.6 Operational Use
  - 4.7 Storage of Keying Material
  - 4.8 Key Update
  - 4.9 Key Recovery
  - 4.10 Key De-registration and Destruction
  - 4.11 Key Revocation

Key Management – Key States & Phases



Key Management – Why

- Two imperatives:
  - Keep keys secure
  - Keep keys available
- Access to the keys means access to the data
- Loss of access to the keys means the data is inaccessible

Key Management – What do you need?

- Key Management system components:
  - Key Generation
    - Needs to support required algorithms
  - Key Distribution
    - Via secure channels
  - Key Material Storage
    - In a secure store, in one or more parts
  - Access Control
    - Control over access to the system itself
    - Control over who/what can use the Key Material
  - Audit
    - Both use of the management system and the Key Material
  - Key Recovery

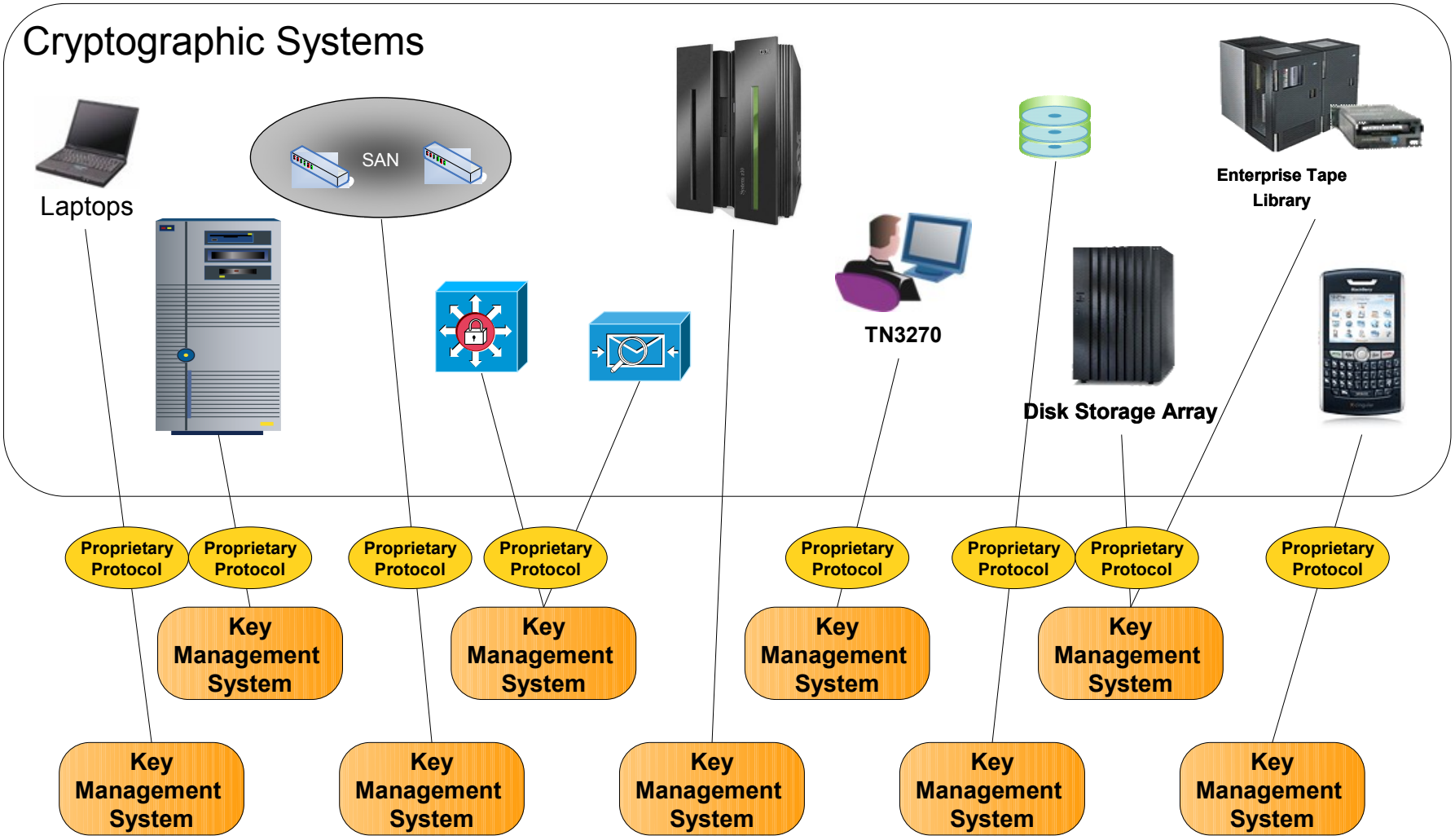
Some things to consider

- "Cryptoperiod" of keys
  - How long should they last before compromise possible?
  - What do you do when the period expires?
- What happens when a key is compromised?
  - Plan for this eventuality
- Encryption deadlock
  - key management data stored on encrypted disk
  - system IPL data on encrypted disk
- Encrypted backup
  - key management data backed up to encrypted tapes!
  - plan as for DR

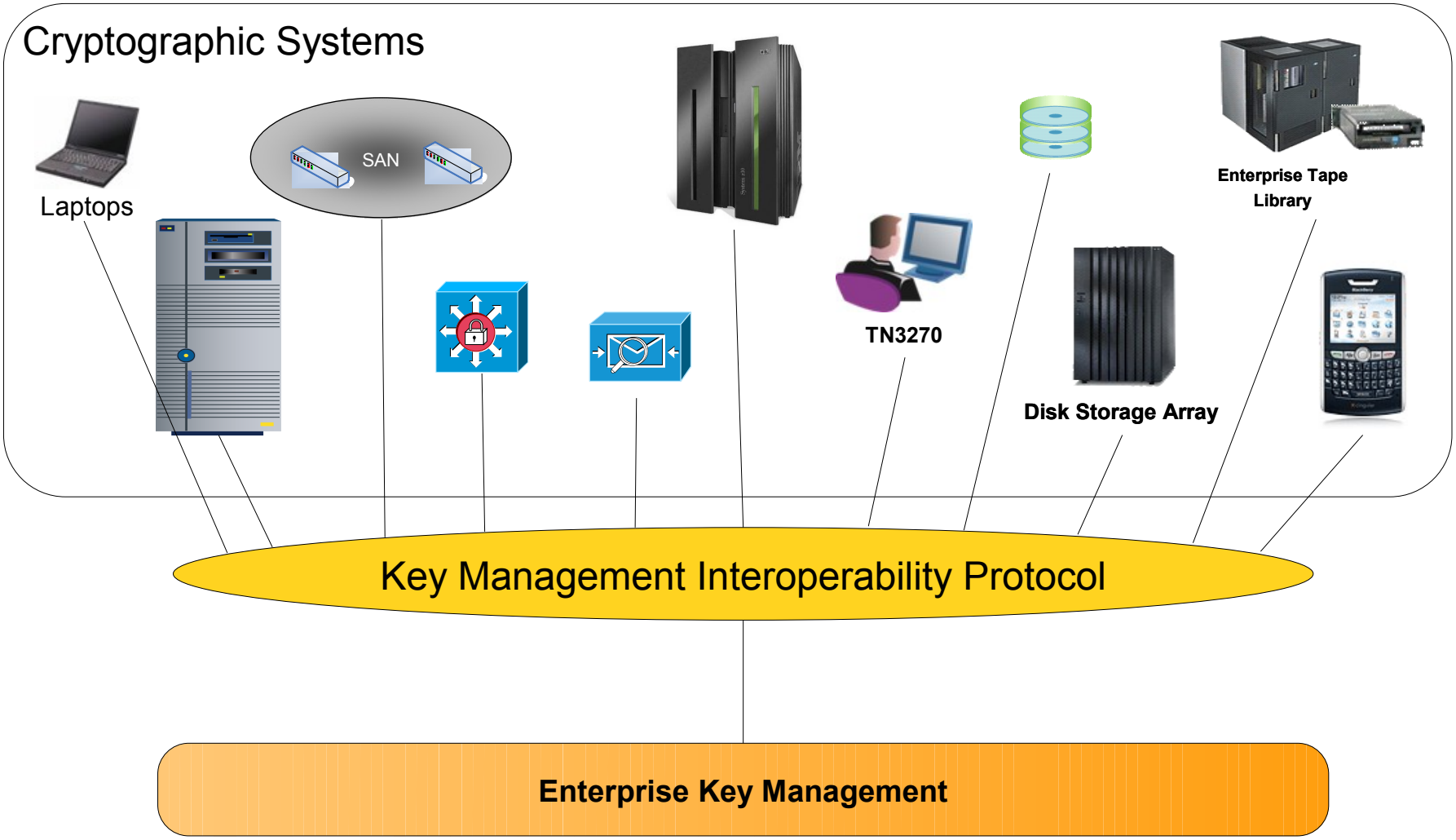
Some best practices

- Security
  - Physical security of hardware/software of key management system
  - Network security for key management operations
  - Access controls for key management
    - Separation of duties
    - Roles
- Availability
  - Key servers should auto-start when power available
  - Network redundancy
- Deadlock prevention
  - Plan, plan, plan
  - Change management
  - Automated monitoring
  - Backup
  - No key deletion?

Key Management – The Conundrum



Key Management – The Holy Grail!



Key Management – and the PCI-DSS

3.5.1 Restrict access to keys to the fewest number of custodians necessary

3.5.2 Store keys securely in the fewest possible locations and forms

3.6.1 Generation of strong keys

3.6.2 Secure key distribution

3.6.3 Secure key storage

3.6.4 Periodic changing of keys

3.6.5 Destruction of old keys

3.6.6 Split knowledge and establishment of dual control of keys

3.6.7 Prevention of unauthorized substitution of keys

3.6.8 Replacement of known or suspected compromised keys

3.6.9 Revocation of old or invalid keys

Summary

- Encrypted does not necessarily mean secured
- Understand WHY you need the data encrypted
  - Use appropriate encryption where possible
- Key management is essential
  - Consider an enterprise approach
  - Get a Key Management Policy
- Yes, PCI are involved too

Acknowledgements, References

- Lennie Dymoke-Bradshaw, IBM
  - for things cryptographic on and off System z

- National Institute of Standards and Technology
  - Recommendation for Key Management, Parts 1 to 3:
    - http://bit.ly/NIST-SP800-57-1 (2nd edition)
    - http://bit.ly/NIST-SP800-57-2
    - http://bit.ly/NIST-SP800-57-3 (draft)
- Key Management Interoperability Protocol
  - Oasis Technical Committee: http://www.oasis-open.org/committees/kmip/
  - White paper - http://bit.ly/KMIP-White