Introduction to Cryptography and Security Mechanisms:

Unit 14

Key Management

Learning Outcomes

- Identify some fundamental principles of key management
- Explain the main phases in the lifecycle of a cryptographic key
- Discuss a number of different techniques for implementing the different phases in the key lifecycle
- Identify appropriate key management techniques for specific application environments
- Appreciate the need for secure key management policies, practices and procedures

Sections

1. Key management fundamentals
2. Key generation
3. Key establishment
4. Key storage
5. Key usage
6. Governing key management

1. Key management fundamentals

What is key management?

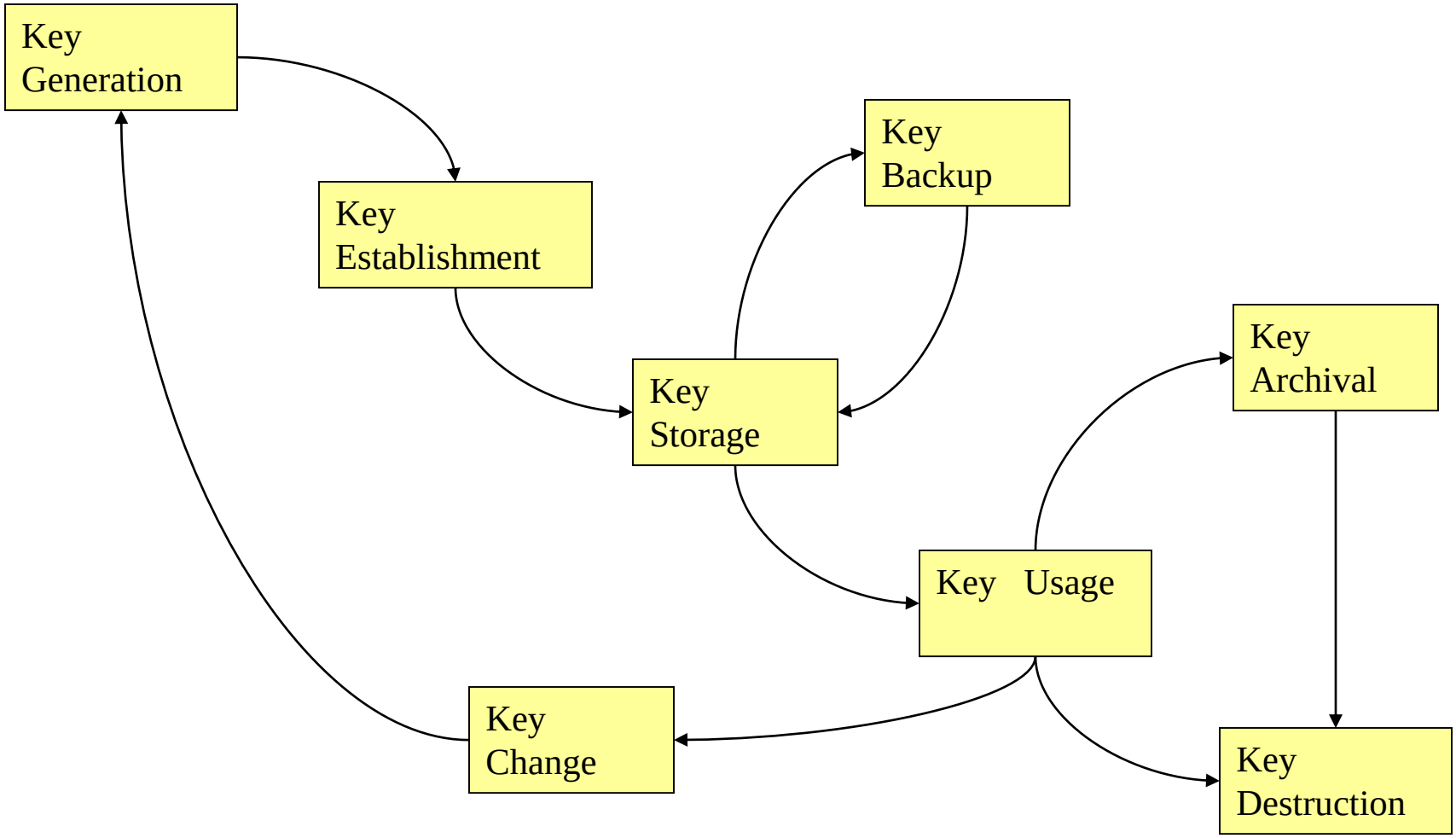
Key management is the secure administration of cryptographic keys

Keys are just special pieces of data!!

What is key management?

- Technical controls
  - e.g. special hardware devices
- Process controls
  - e.g. dealing with lost keys
- Environmental controls
  - controls depends on physical location
- Human factors

Lifecycle of a cryptographic key



Fundamental requirements

- Secrecy of secret keys
  - symmetric keys
  - private keys

- **Assurance of purpose** of keys
  - Someone in possession of a key should be confident that they can use the key for the purpose that they believe it to be for (this includes integrity of the key itself)

Key management system

A **key management system** is any system for managing the various stages of the key lifecycle.

A KMS needs to be aligned with the functionality and priorities of the supporting organisation and depends on:

- Network topology
- Cryptographic mechanisms
- Compliance restrictions
- Legacy issues

2. Key generation

Key lifetimes

All keys have finite **lifetimes** in order to support:

- Mitigation against key compromise
- Mitigation against key management failure
- Limitation of exposure
- Enforcement of management cycles
- Mitigation against future attacks
- Flexibility

Key lengths

- Listen to the experts!!
  - www.keylength.com
- Key lengths for symmetric cryptography tend to be algorithm-independent
- Key lengths of public key cryptography tend to be algorithm-specific
- Advice on key length changes over time!

Direct symmetric key generation

- Symmetric keys are just random numbers
- Hardware-based non-deterministic generators best, but most expensive…
- Software-based deterministic generators least secure, but cheapest to deploy…

Symmetric key derivation

Key derivation is the generation of cryptographic keys from other cryptographic keys.

$K1 = h(K \| 0)$ $K2 = h(K \| 1)$

- **Efficiency**: key generation and establishment are costly process
- **Longevity**: may want to preserve security of long-term keys

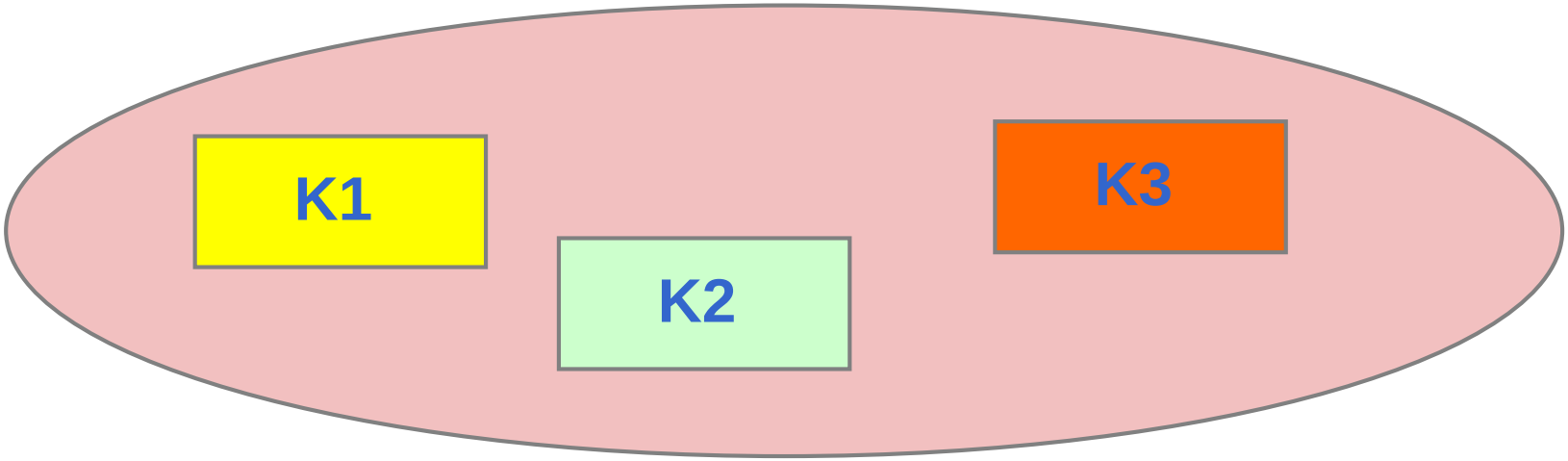
Key derivation from a PIN/password

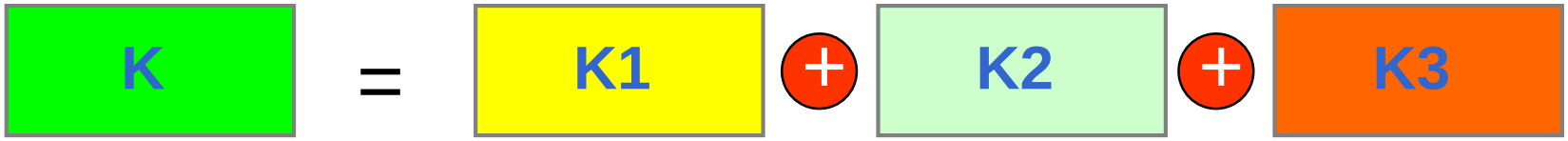
PKCS#5 defines how a key can be derived from a password/PIN

K = F(P,S,C,L), where:

- F: key derivation function
- P: password or PIN
- S: 64 pseudorandom bits
- C: iteration counter
- L: length of derived key

Key generation from components





Asymmetric key pair generation

- Algorithm-specific
- Often requires random numbers
- Normally relatively slow and complex
- Relative standards should be consulted

3. Key establishment

Ease of key establishment

This is normally one of the hardest key management processes, but can be easy when:

- the key does not need to be shared
- the key does not need to be secret
- the key can be established within a controlled environment

Key hierarchies

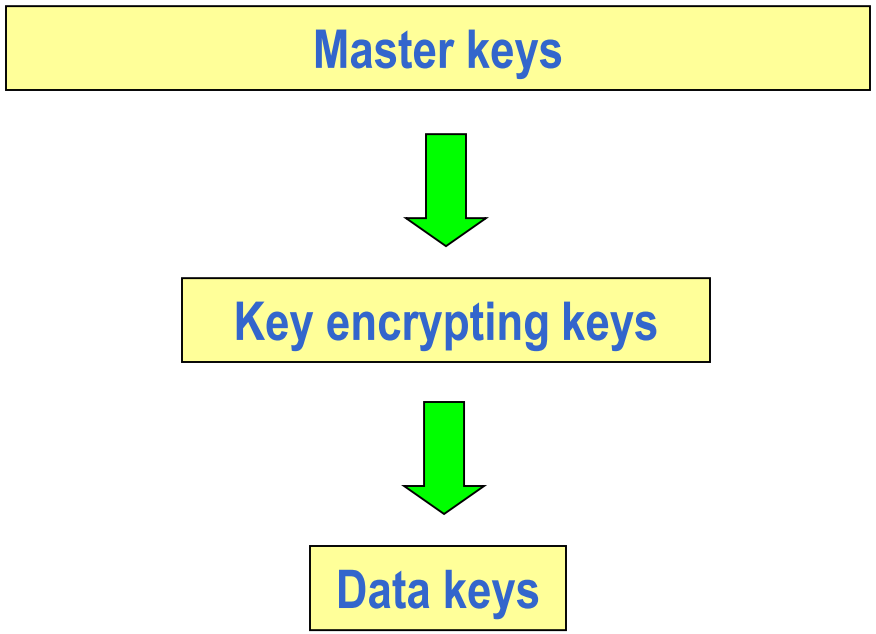
Key hierarchies are widely used in (symmetric) key management systems.

Keys are ranked in importance and “layered”.

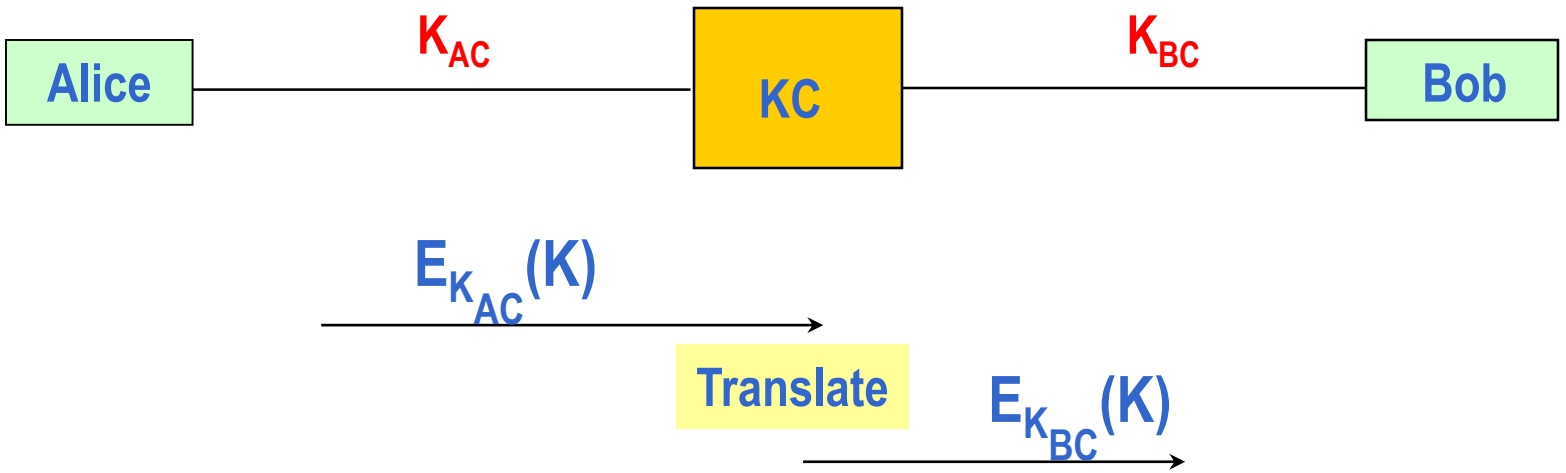
The two main advantages are:

1. Aids secure distribution and storage
2. Limits exposure of important keys

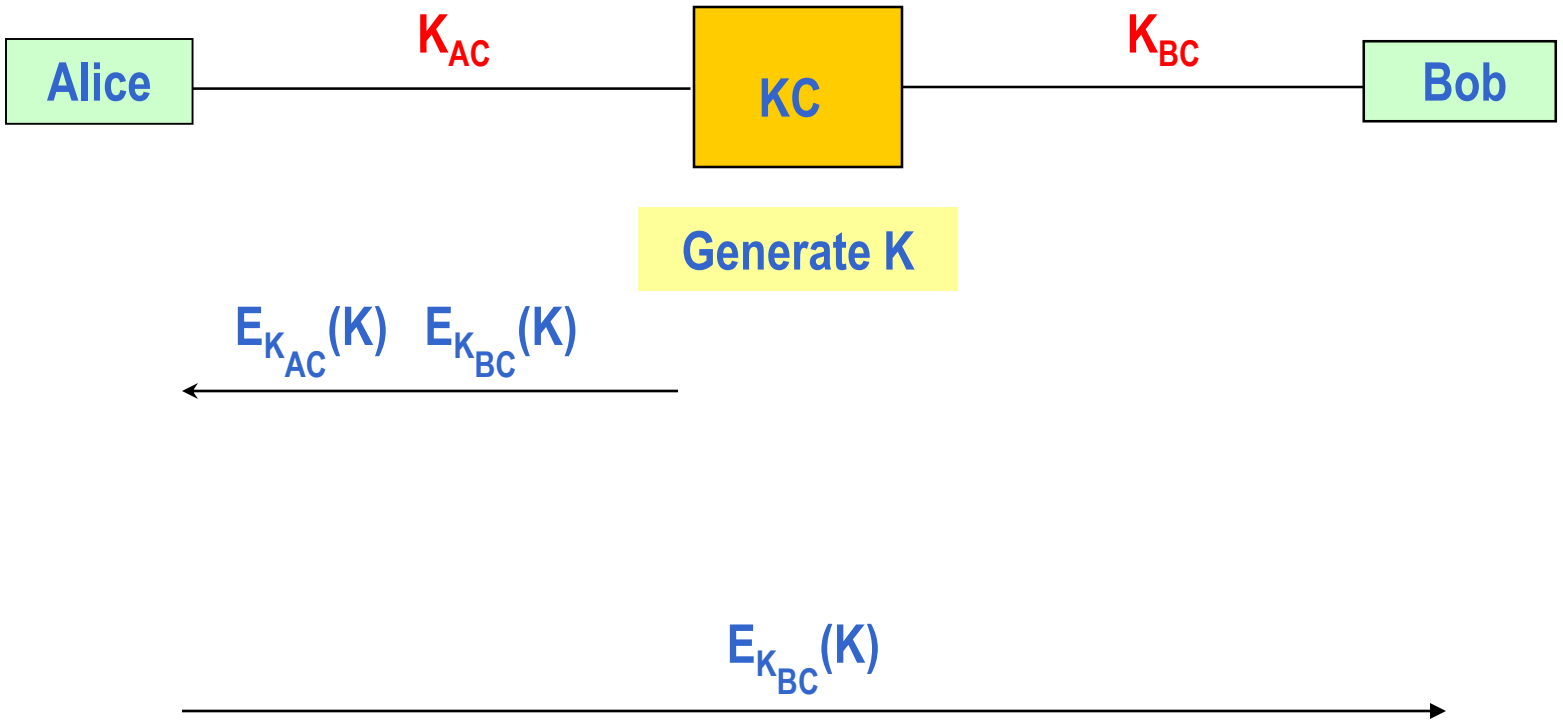
Three-level key hierarchy



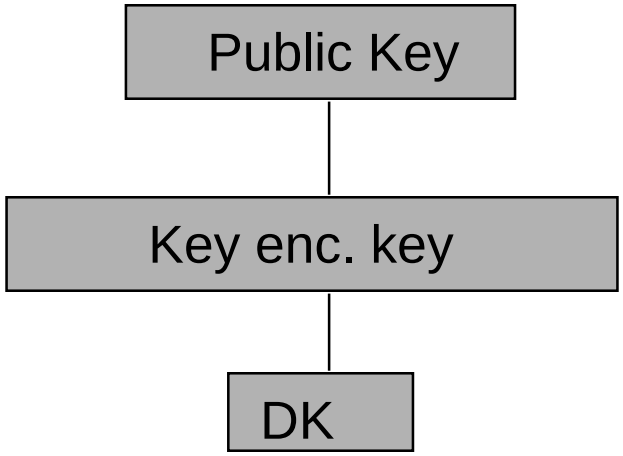
Key translation



Key despatch



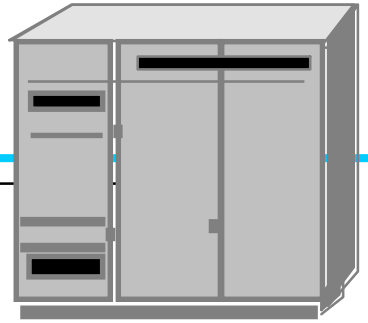
Hybrid key hierarchies



- Particularly useful for many-to-many systems (e.g. SSL).
- Only public keys need to be distributed in advance - no need for secrecy (although assurance of purpose is required).

Unique key per transaction schemes

- By deriving keys from information already known by the parties involved, we can:
  - avoid long-term key storage
  - avoid dedicated key establishment process
  - automate key generation and establishment
- UKPT schemes useful for retail point-of-sale terminals



Host

key register

Derive transaction key K

key register

request || MAC_K(request)

Derive transaction key K

response || MAC_K(response)

Update key register

key register

Update key register

key register

Generic UKPT scheme

Designing a UKPT scheme

- What is the initial value in the key registers?
- How should the transaction key(s) be derived?
- How should the key registers be updated?

Example: Racal UKPT scheme

- What is the initial value in the key registers?
  - *A secret seed agreed between terminal and host*
- How should the transaction key(s) be derived?
  - *Using a key derivation function that takes as input the current key register value and the primary account number of the card*
- How should the key registers be updated?
  - *Using a key derivation function that takes as input the current key register value, the primary account number of the card, and the (not transmitted) MAC residues of the request and response messages*

Quantum key establishment

- If we could make key establishment “easy”, we could use the one-time pad…
- QKE is a technique for using quantum mechanisms to establish a conventional symmetric key
- QKE is not “quantum cryptography”
- Allows Alice and Bob to exchange data “in the clear” in such a way that they can test whether an attacker has been listening to the channel

BB84 protocol

- Alice generates stream of qubits and sends these as a stream of polarised photons to Bob
- Bob measures them using a polarisation detector
- Bob contacts Alice over a conventional authentication channel:
  - Alice confirms which detector Bob should have used for each measurement – results in 50% of the bits being discarded
  - Alice and Bob conduct random checks on remaining bits in order to detect tampering

QKE in practice

- Distance limitations
  - around 300kms possible in 2015 via optical fibre
  - not currently believed to work beyond 400kms
- Data rates
  - limited and deteriorate for longer ranges
- Cost
  - requires special hardware
- Need for conventional authentication

4. Key storage

Key storage in software

- Hidden in software, “in the clear”
  - Cheap but dangerous!
  - Developer who designs software knows where they are
  - An attacker who has access to two different versions (with different keys) can run a comparison
- Storing key in encrypted form
  - So where do we store the “key encrypting key”?

Key storage in hardware

- Can utilise tamper-resistant features that may be used include:

  Micro-switches
  Electronic mesh
  Potting sensitive components in resin
  Temperature detectors
  Light-sensitive diodes
  Movement / tilt detectors
  Voltage / current detectors
  Secure chips

- Choice of features depends on location of hardware modules:
  - e.g. physically secure environments (computer centre)
  - e.g. retail environment

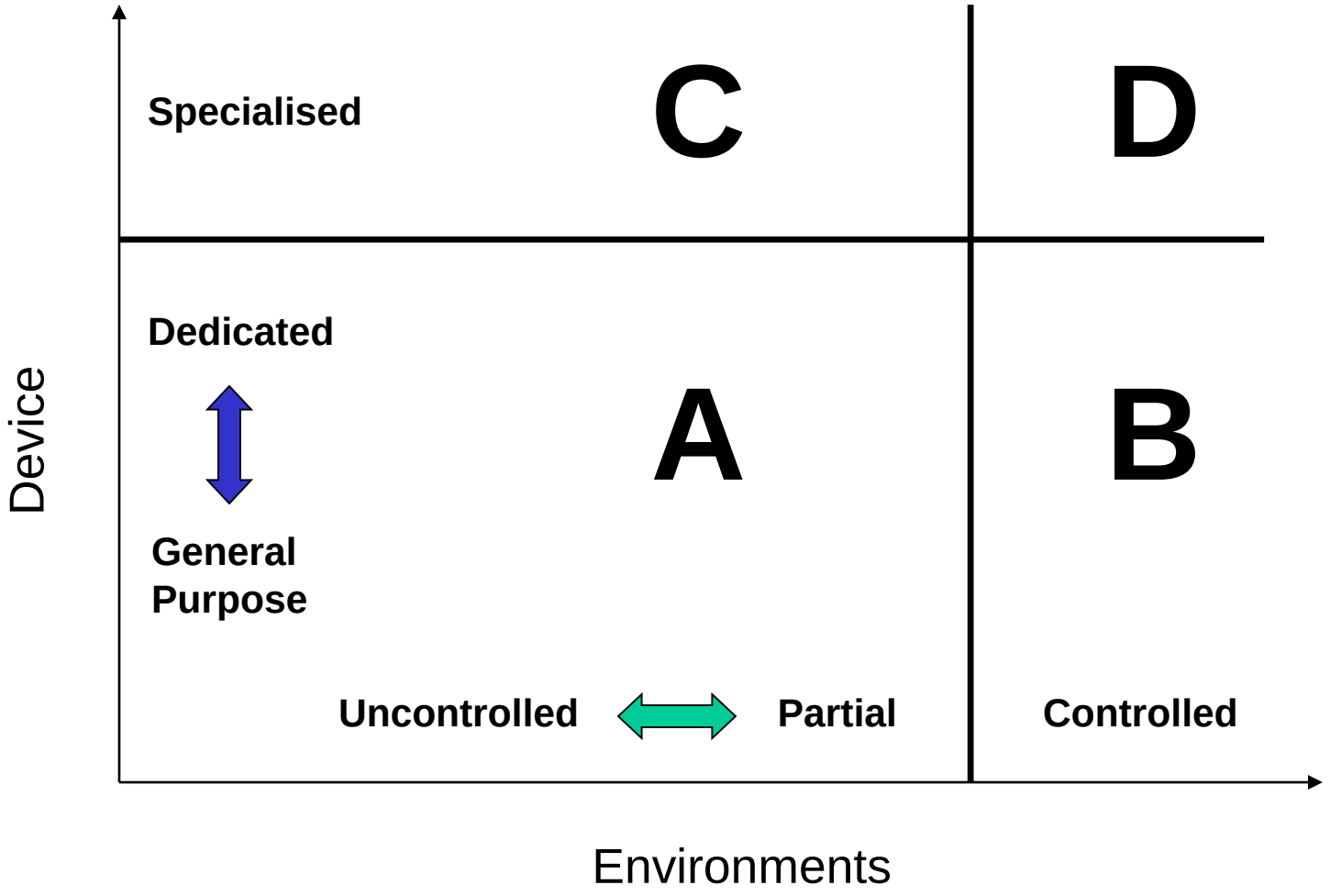
Hardware Security Modules

- Dedicated hardware devices that provide key management functionality
- Use a combination of the previous techniques to provide physical protection
- Backed up by a battery
- At least one key resides in the HSM at all times
- All other keys only leave the HSM in encrypted form

Other hardware issues

- Relies on a secure communication with hardware
  - Have been attacks based on manipulating the application programming interface
- HSMs are critical security components
  - Need to be evaluated by specialists and benchmarked against appropriate standards

Key storage risk factors



Other key storage issues

- Key backup
- Key archival
- Key recovery

5. Key usage

Principle of key separation

A cryptographic key should only be used for its intended purpose

Why key separation?

- Different cryptographic services may require different key lengths
- Different exposure levels
  - e.g. not using a master key as a data key
- Unidirectional keys
  - A PIN encryption key should not be used as a data key and vice-versa

Enforcing key separation

- Encrypting a key using a specified variant key and enforcing in an HSM

- Embedding a key in a larger data block
  - Employ redundancy (key tagging)
  - Key blocks
  - Public key certificates

Key blocks

- ANSI has defined a TR-31 key block, to ensure that a key can only be used for its intended purpose.
- The key block may be usable for either key storage or key distribution.

- Header includes key usage, mode of use, exportability, algorithm.
- Key encrypted using a variant of the storage or distribution key, in CBC mode.
- Authenticator calculated using a different variant of the storage/distribution key.
- Currently, only DES and 3-DES supported.

Key separation in practice

- Key separation is a **principle**
- Great care should be taken if using a key for different purposes
- Key derivation a pragmatic compromise to reusing a key

Reasons for key change

- Planned key updates
  - end of key lifetime
- Unplanned key updates
  - key compromise
  - security vulnerability discovered that could lead to key compromise
  - unexpected departure of an employee

Impact of key change

- Minimum impact is the need for a key new to be generated and established

- Impact could include:
  - costs of distributing new smart cards
  - costs of investigation
  - costs of changing key management system
  - costs relating to repairing activities conducted using compromised keys
  - damage to reputation
  - loss of confidence

Key activation

Key activation is the process by which the use of a key is “authorised”:

Key activation often uses a less secure mechanism than suggested by the protection offered by the key, for example:

- by entering a password
- by accessing or connecting a device

Key destruction

- Must be done using a secure mechanism for destruction of sensitive data
  - when key expires
  - when key is withdrawn
  - at end of a key archival period
- Usually involves over-writing memory, but may involve physical destruction

6. Governing key management

Policies, practices and procedures

- Key management policies
  - Define overall strategy
- Key management practices
  - Tactics used to achieve the policy
- Key management procedures
  - Step-by-step tasks to implement practices

Key Management Advice

There are many standards and guidance relating to key management.

For example:

ANSI X9.17 / ISO 8732 (withdrawn)
ANSI X9.24 Retail Financial Services Symmetric Key Man.
APACS 40 & APACS 70 (UK)
ISO 11568 Key management (retail)
ISO 11770 Security techniques – key management
ISO 15782 Certificate management (financial services)
RFC 4107 Guidelines for cryptographic key management
NIST SP 800-57 Recommendations for key management
IEEE P1619.3 Key management standard working group
OASIS Key Management Interoperability Protocol

Summary

- Key management is the aspect of cryptography of greatest relevance to users, decision-makers and implementers of cryptography
- The entire lifecycle of a cryptographic key must be kept in mind when designing a key management system
- Key management involves controls at many different layers (technical, physical, human) and requires careful governance