Data Set Level Encryption

Key Management

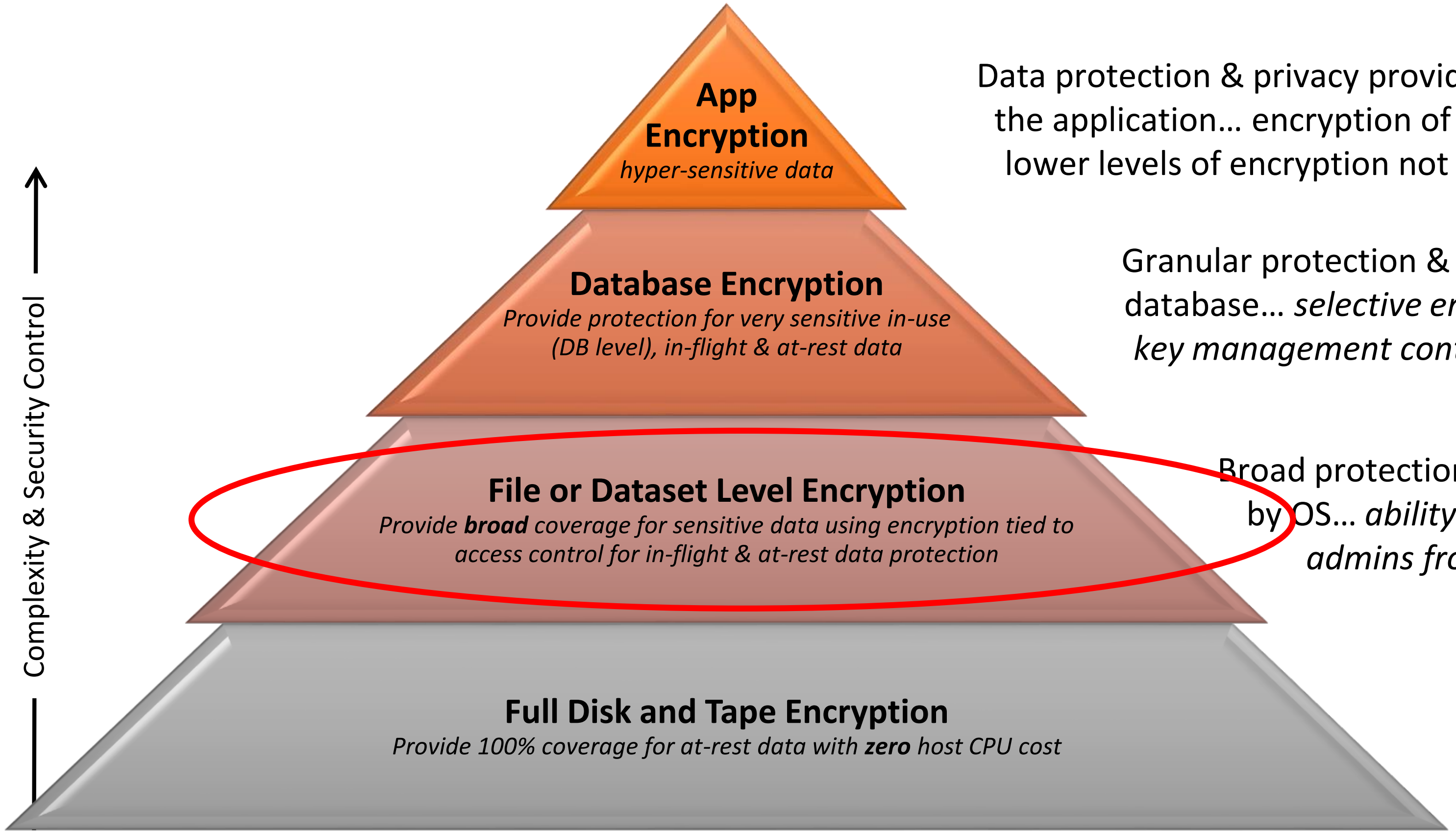
Heinz Tschumi, IBM Switzerland
htsi@ch.ibm.com

Data Set Level Encryption

Key Management

Introduction

Multiple layers of encryption for Data Sets



App Encryption
hyper-sensitive data

Data protection & privacy provided and managed by the application… encryption of sensitive data when lower levels of encryption not available or suitable

Database Encryption
Provide protection for very sensitive in-use (DB level), in-flight & at-rest data

Granular protection & privacy managed by database… *selective encryption & granular key management control of sensitive data*

File or Dataset Level Encryption
*Provide **broad** coverage for sensitive data using encryption tied to access control for in-flight & at-rest data protection*

Broad protection & privacy managed by OS… *ability to eliminate storage admins from compliance scope*

Full Disk and Tape Encryption
*Provide 100% coverage for at-rest data with **zero** host CPU cost*

Protection against intrusion, tamper or removal of *physical* infrastructure

Clear Key / Secure Key / Protected Key

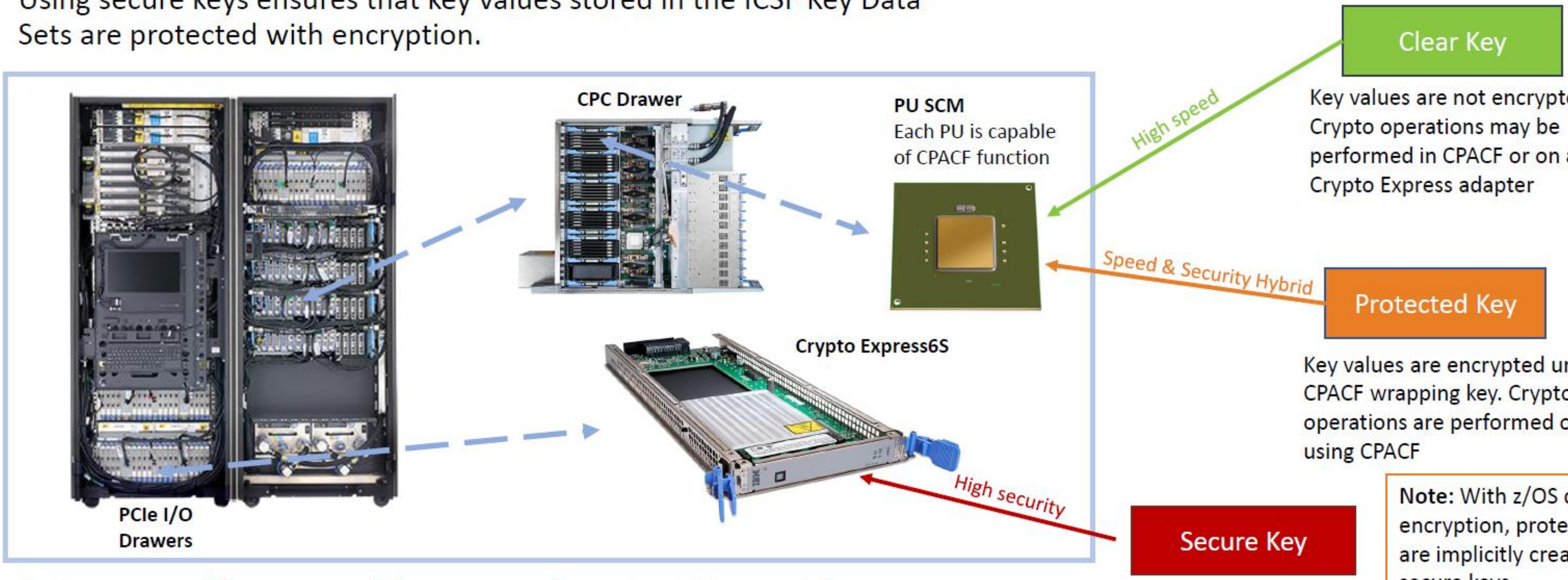
Clear Key – key may be in clear, at least briefly, somewhere in the operating system

Secure Key – key value does not exist in the clear outside of the HSM (Crypto Express Card)

Fort Knox

Protected Key – key value does not exist outside of physical hardware (HSA)

Using secure keys ensures that key values stored in the ICSF Key Data Sets are protected with encryption.



Clear Key

Key values are not encrypted. Crypto operations may be performed in CPACF or on a Crypto Express adapter

Protected Key

Key values are encrypted under a CPACF wrapping key. Crypto operations are performed only using CPACF

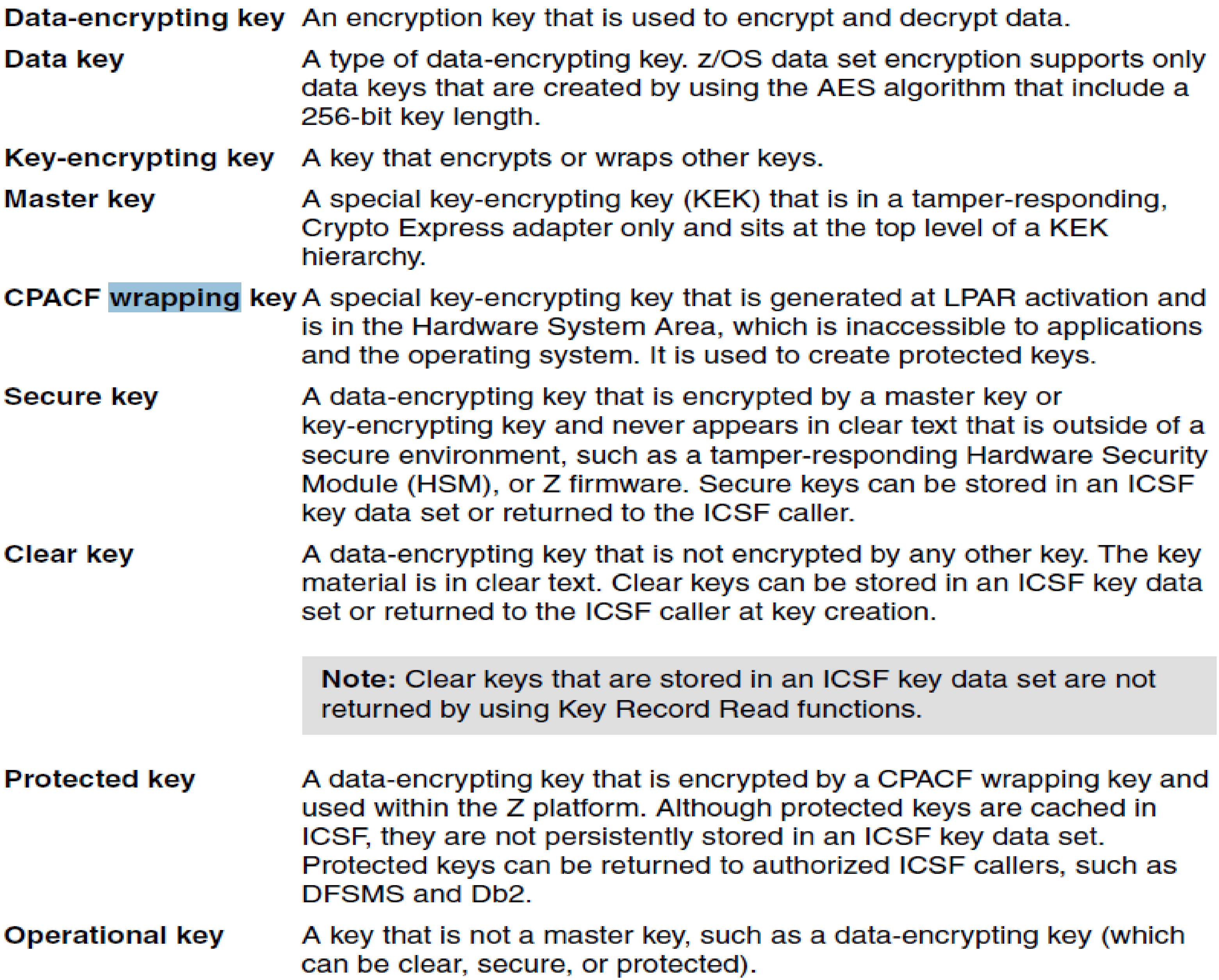
Secure Key

Key values are encrypted under a Master Key. Crypto operations are performed only on a Crypto Express adapter

Note: With z/OS data set encryption, protected keys are implicitly created from secure keys.

Only protected keys created from secure keys should be used for Pervasive Encryption.

Important Terms



Data-encrypting key An encryption key that is used to encrypt and decrypt data.

Data key A type of data-encrypting key. z/OS data set encryption supports only data keys that are created by using the AES algorithm that include a 256-bit key length.

Key-encrypting key A key that encrypts or wraps other keys.

Master key A special key-encrypting key (KEK) that is in a tamper-responding, Crypto Express adapter only and sits at the top level of a KEK hierarchy.

CPACF wrapping key A special key-encrypting key that is generated at LPAR activation and is in the Hardware System Area, which is inaccessible to applications and the operating system. It is used to create protected keys.

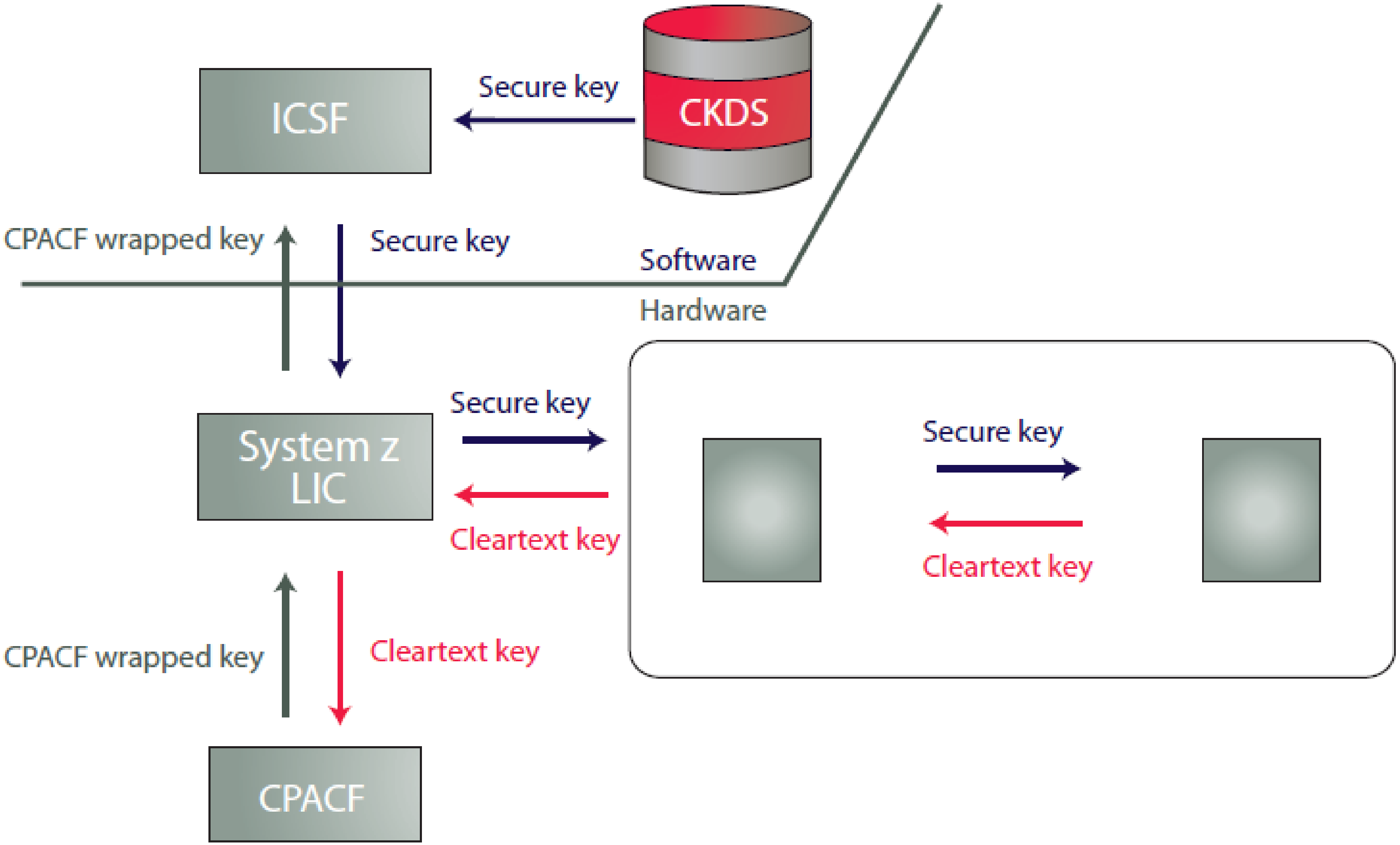
Secure key A data-encrypting key that is encrypted by a master key or key-encrypting key and never appears in clear text that is outside of a secure environment, such as a tamper-responding Hardware Security Module (HSM), or Z firmware. Secure keys can be stored in an ICSF key data set or returned to the ICSF caller.

Clear key A data-encrypting key that is not encrypted by any other key. The key material is in clear text. Clear keys can be stored in an ICSF key data set or returned to the ICSF caller at key creation.

> **Note:** Clear keys that are stored in an ICSF key data set are not returned by using Key Record Read functions.

Protected key A data-encrypting key that is encrypted by a CPACF wrapping key and used within the Z platform. Although protected keys are cached in ICSF, they are not persistently stored in an ICSF key data set. Protected keys can be returned to authorized ICSF callers, such as DFSMS and Db2.

Operational key A key that is not a master key, such as a data-encrypting key (which can be clear, secure, or protected).



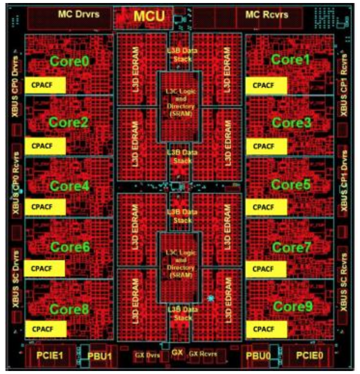


Figure 11. Transforming a CCA-encrypted key token into a CPACF-wrapped key

➢ CPACF Wrapping Key

- Generated at LPAR Activation
- Resides in the HSA (Hardware Systems Area) in a protected area
- Is not visible to Operating System or Applications
- **SYMCPACFWRAP (YES ¦ NO)** specifies whether symmetric keys can be rewrapped by CPACF

➢ Protected Key

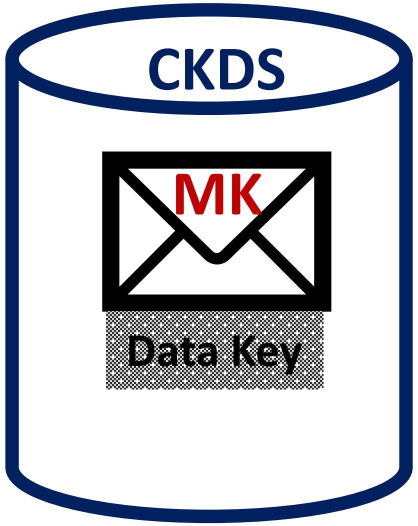
- For high speed encryption
- Generated from a secure key
- Not stored in CKDS / stored in Memory (ICSF address space) only
- Never in clear available for Operating System and/or Applications

Data Set Level Encryption

Key Management

From Secure Key to Protected Key

ICSF



SW

CKDS = Cryptographic Key Data Set

HW

HSA

CPACF

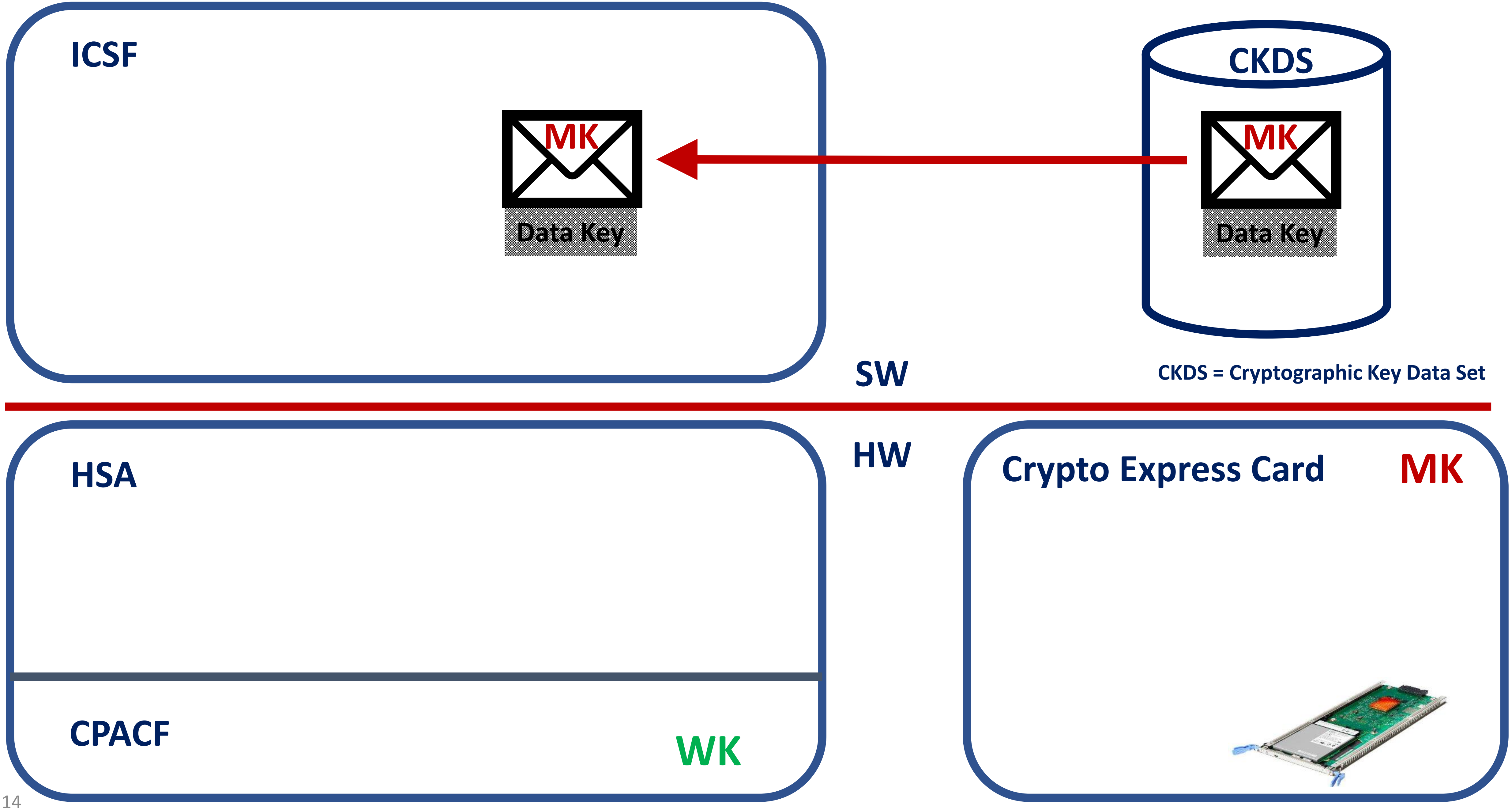
WK = CPACF Wrapping Key

WK

Crypto Express Card

MK

MK = Master Key



ICSF

MK

Data Key

CKDS

MK

Data Key

SW

CKDS = Cryptographic Key Data Set

HW

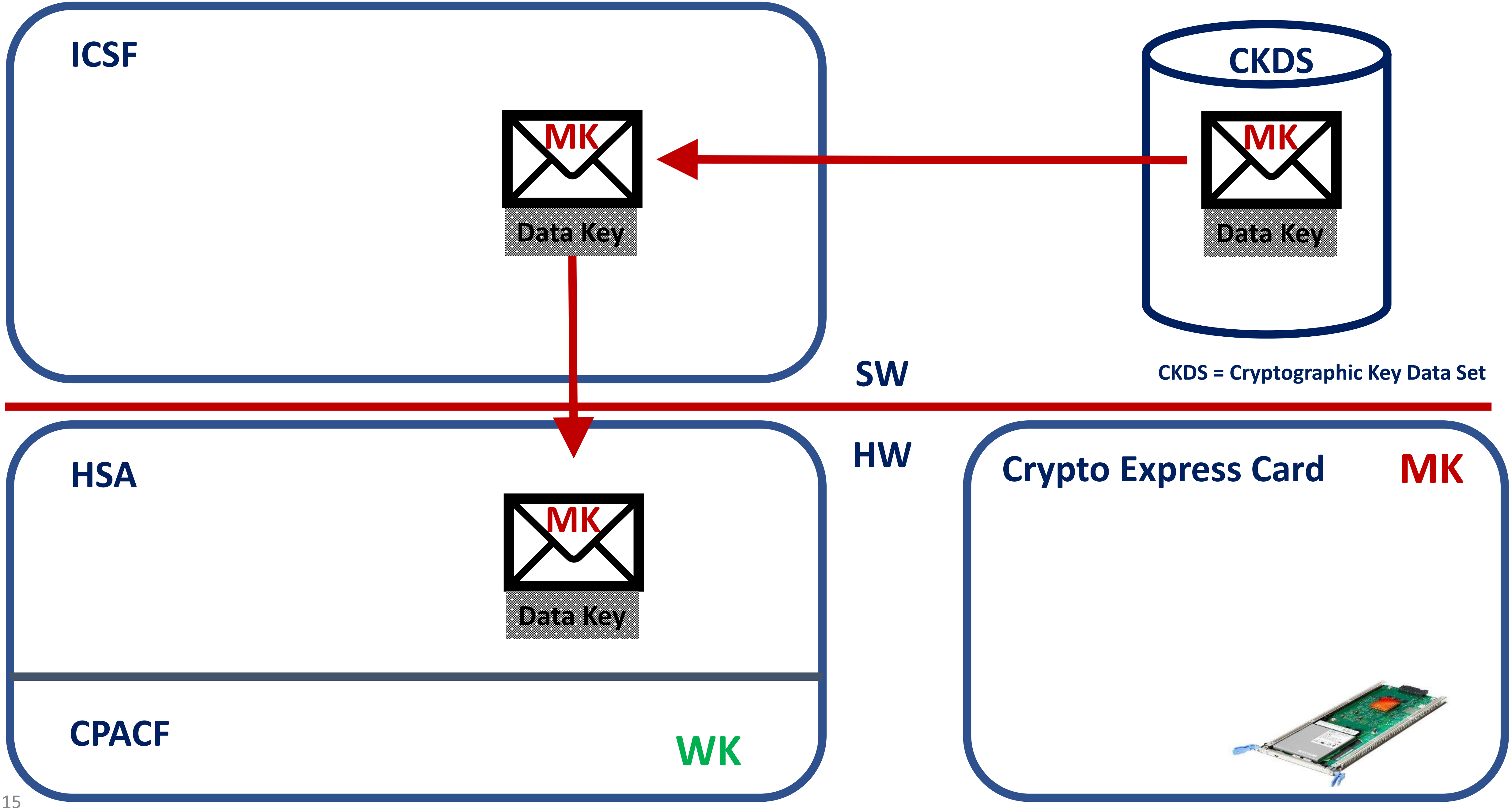
HSA

CPACF

WK

Crypto Express Card

MK



ICSF

MK

Data Key

CKDS

MK

Data Key

SW

CKDS = Cryptographic Key Data Set

HW

HSA

MK

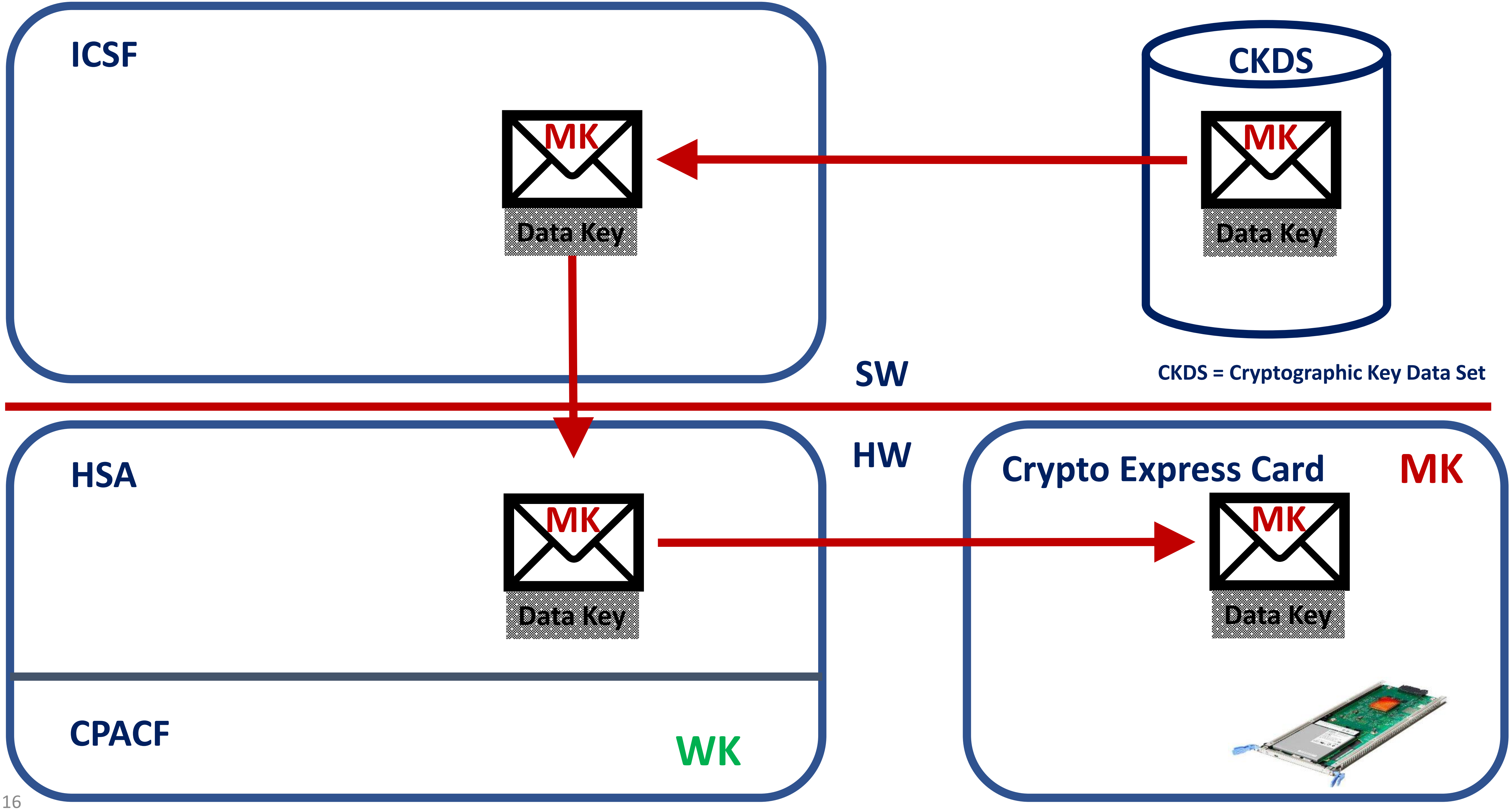
Data Key

CPACF

WK

Crypto Express Card

MK



ICSF

MK

Data Key

CKDS

MK

Data Key

SW

CKDS = Cryptographic Key Data Set

HW

HSA

MK

Data Key

CPACF

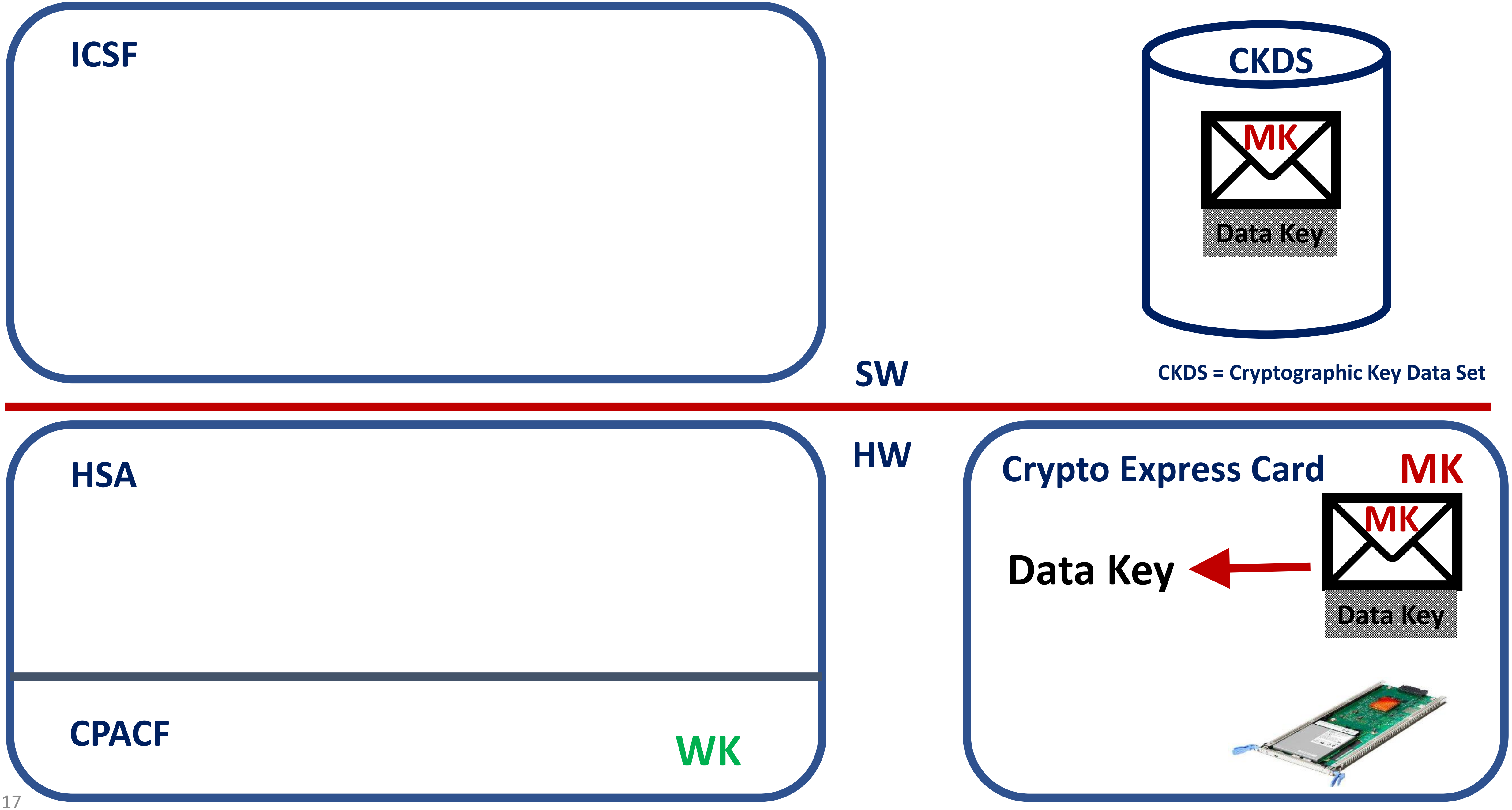
WK

Crypto Express Card

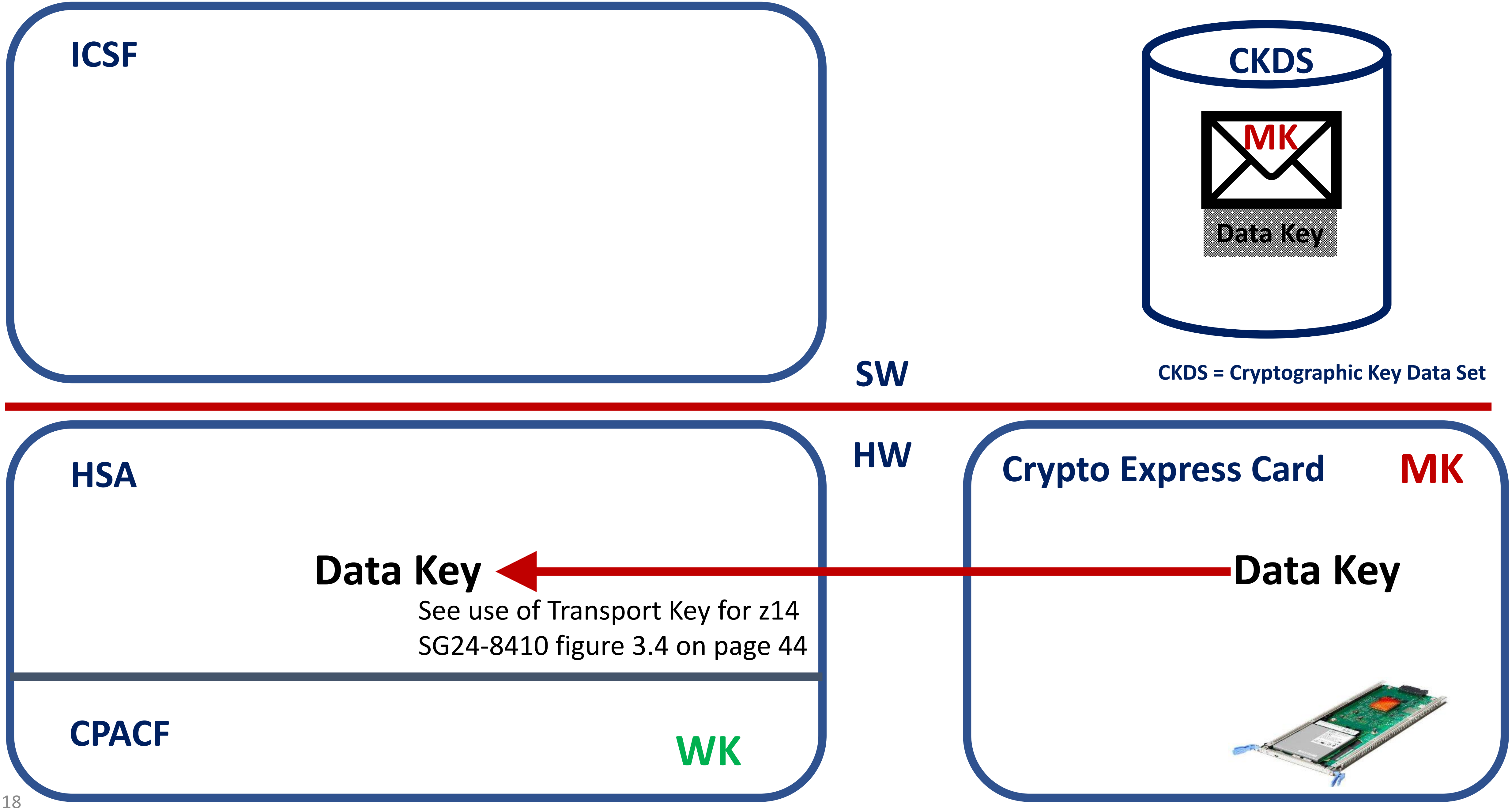
MK

MK

Data Key



CKDS = Cryptographic Key Data Set



ICSF

CKDS

MK

Data Key

CKDS = Cryptographic Key Data Set

SW

HW

HSA

Data Key

See use of Transport Key for z14
SG24-8410 figure 3.4 on page 44

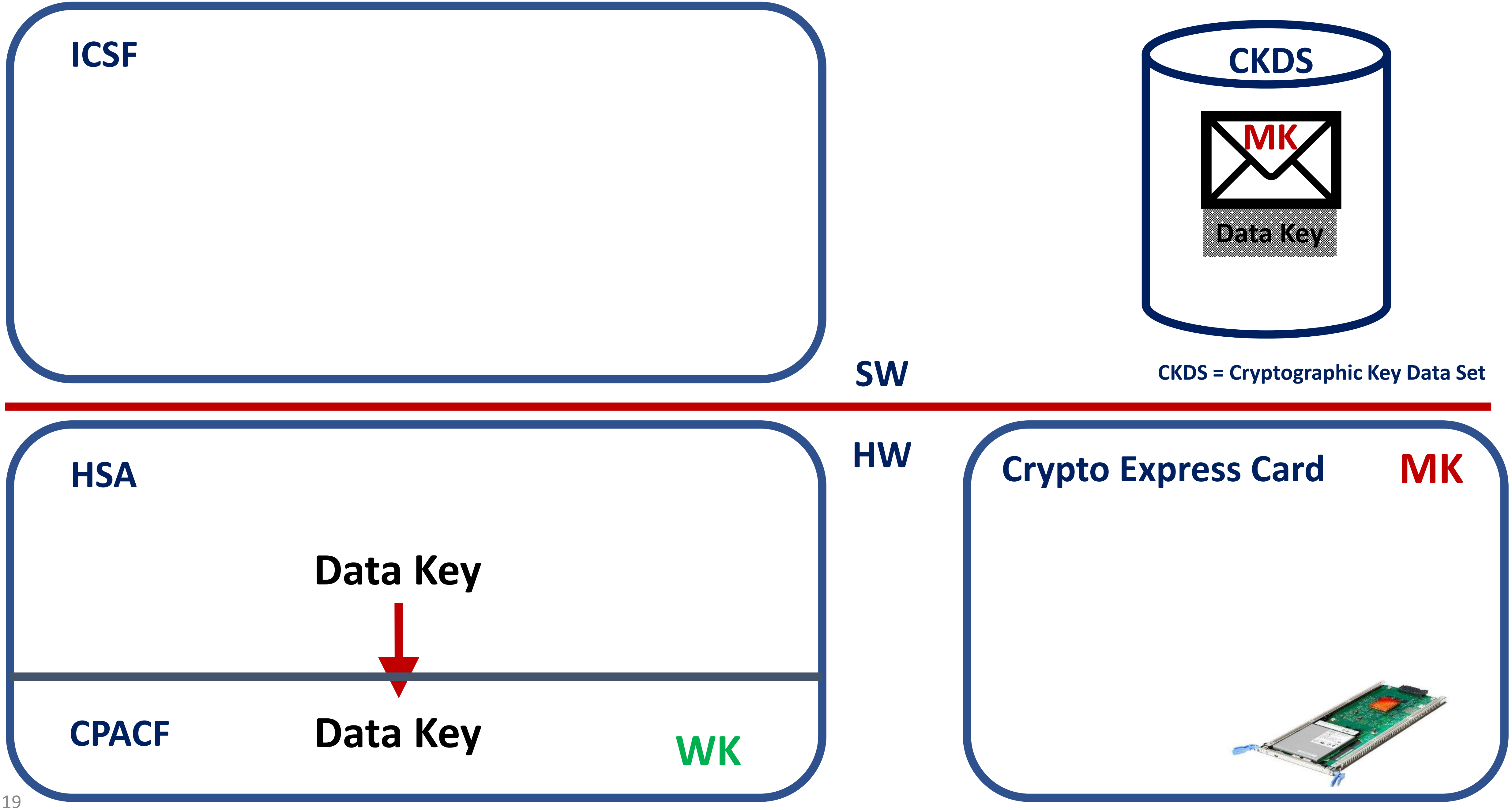
CPACF

WK

Crypto Express Card

MK

Data Key



ICSF

CKDS

MK

Data Key

SW

CKDS = Cryptographic Key Data Set

HW

HSA

Data Key

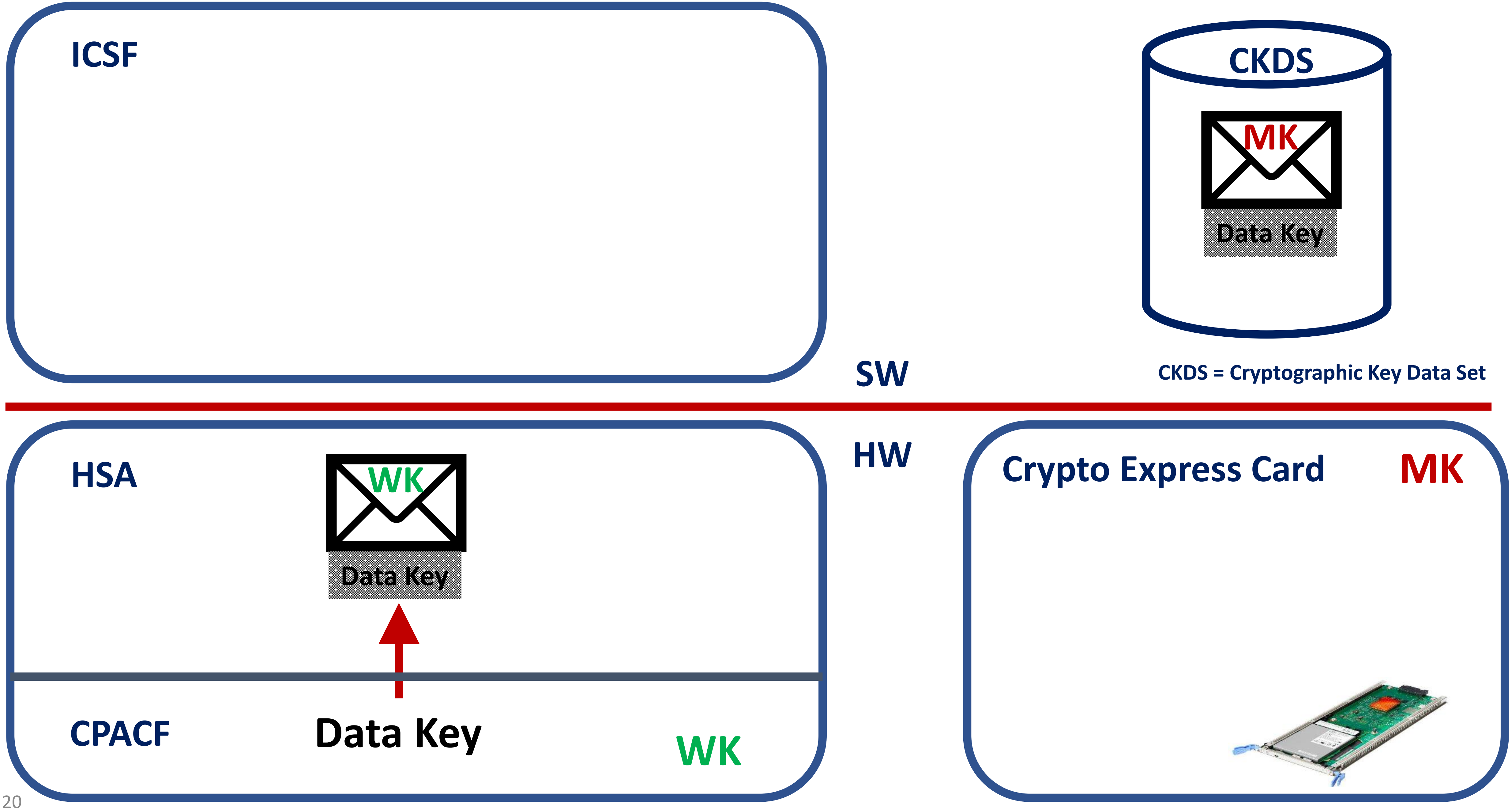
CPACF

Data Key

WK

Crypto Express Card

MK



ICSF

CKDS

MK

Data Key

SW

CKDS = Cryptographic Key Data Set

HW

HSA

WK

Data Key

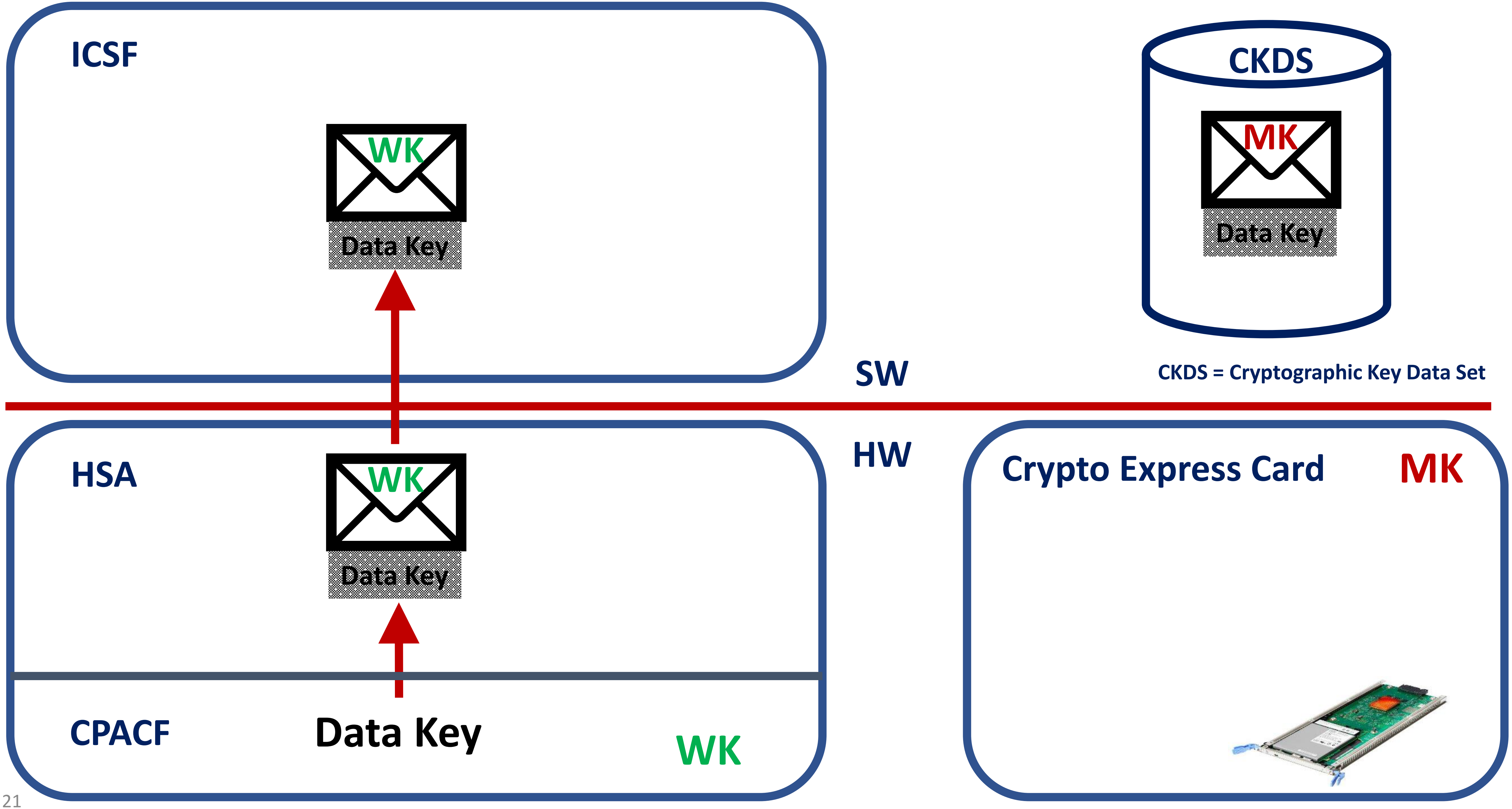
CPACF

Data Key

WK

Crypto Express Card

MK



ICSF

WK

Data Key

CKDS

MK

Data Key

SW

CKDS = Cryptographic Key Data Set

HW

HSA

WK

Data Key

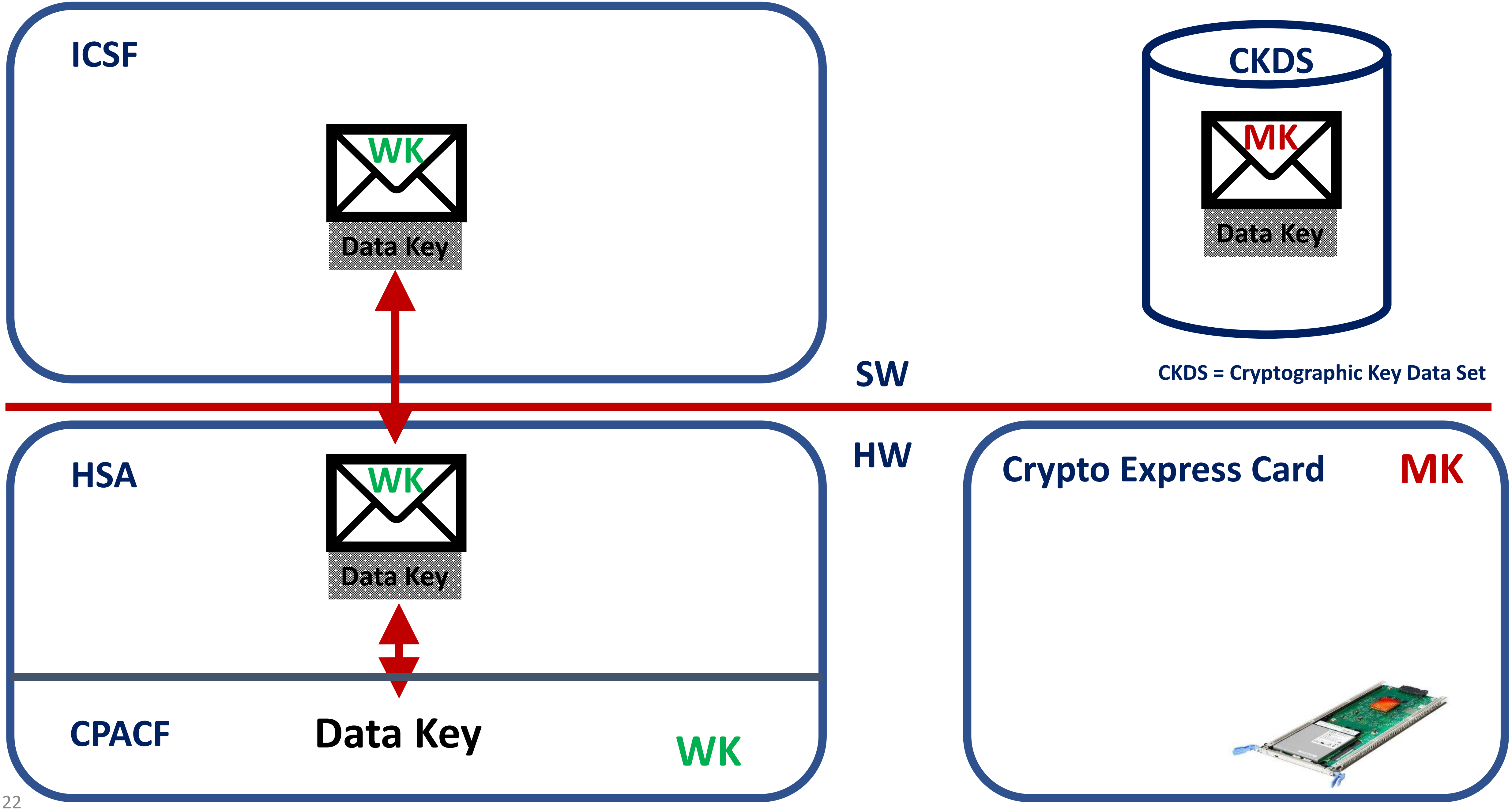
Crypto Express Card

MK

CPACF

Data Key

WK



ICSF

WK

Data Key

CKDS

MK

Data Key

SW

CKDS = Cryptographic Key Data Set

HW

HSA

WK

Data Key

Crypto Express Card

MK

CPACF

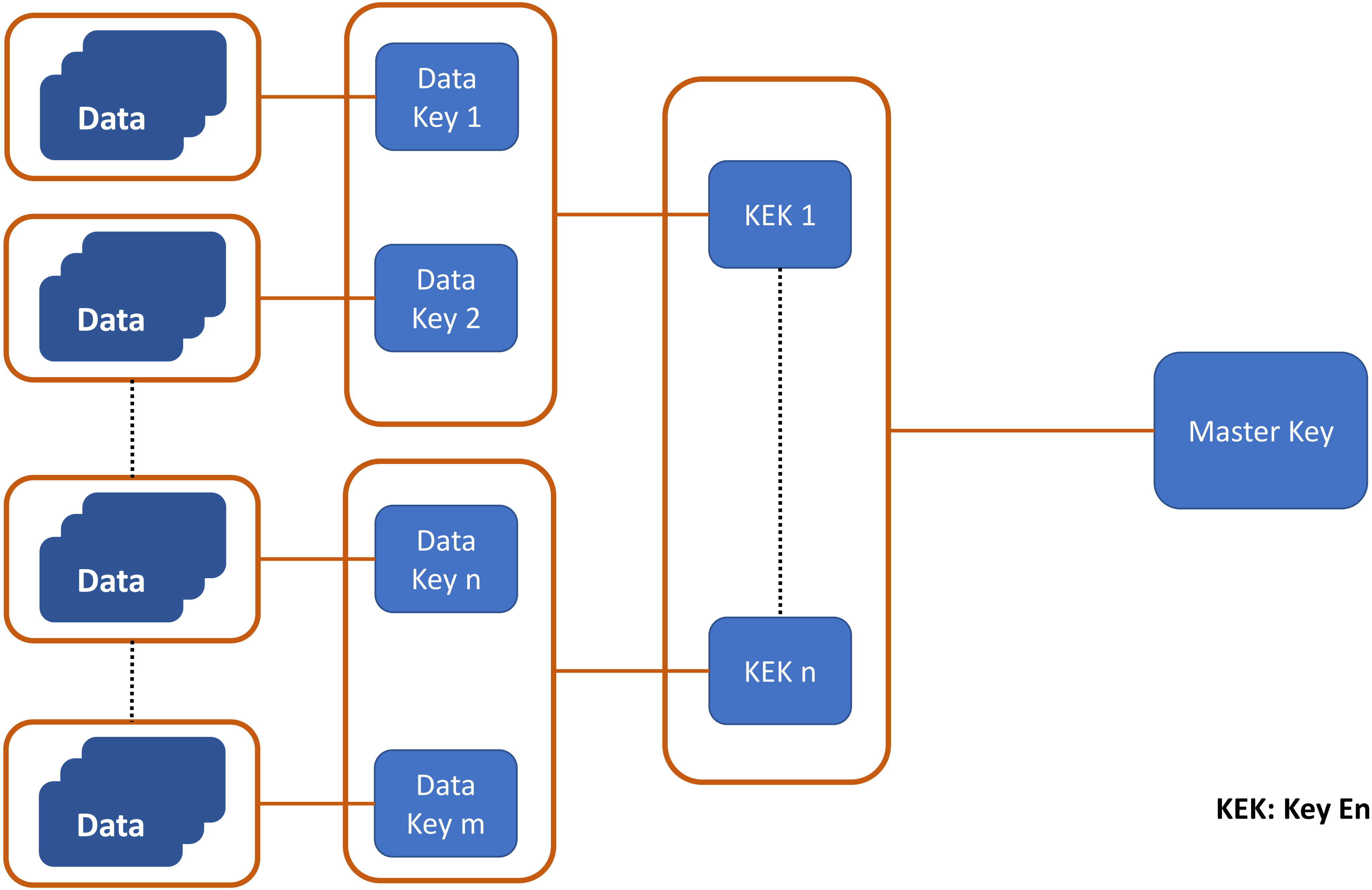
Data Key

WK

Data Set Level Encryption

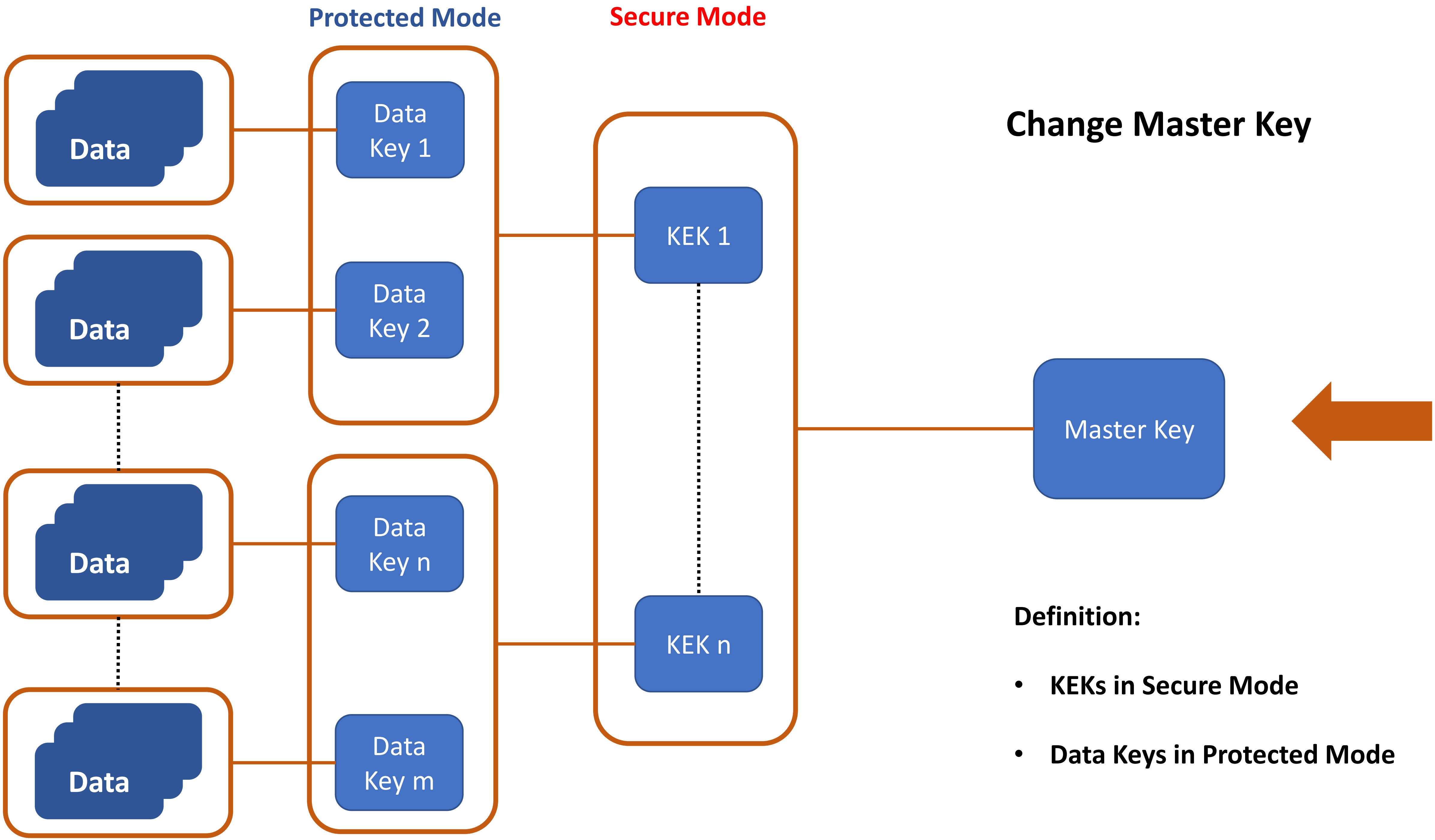
Key Management

Key Change



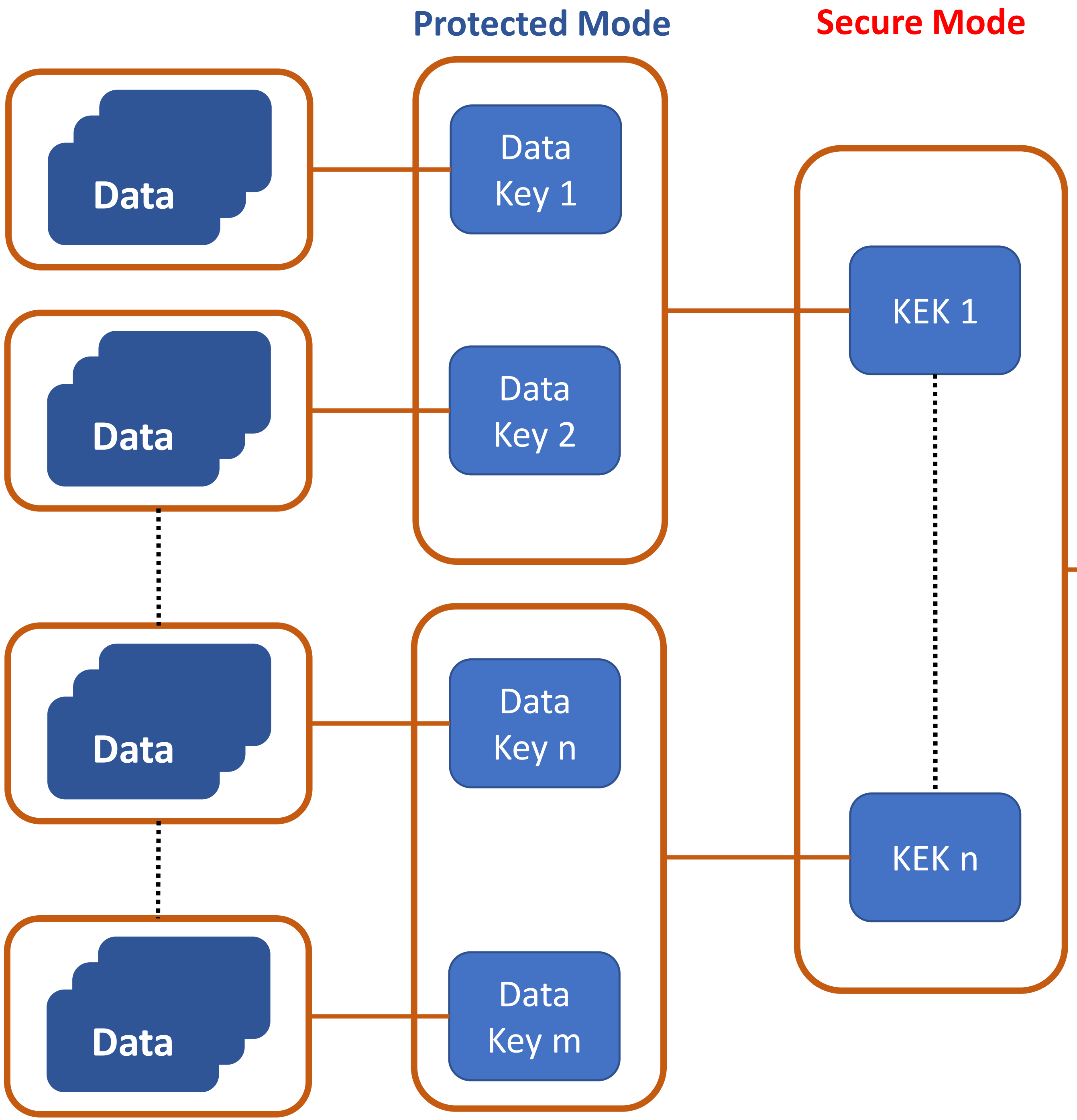
KEK: Key Encryption Key

Change Master Key



Definition:

- **KEKs in Secure Mode**
- **Data Keys in Protected Mode**



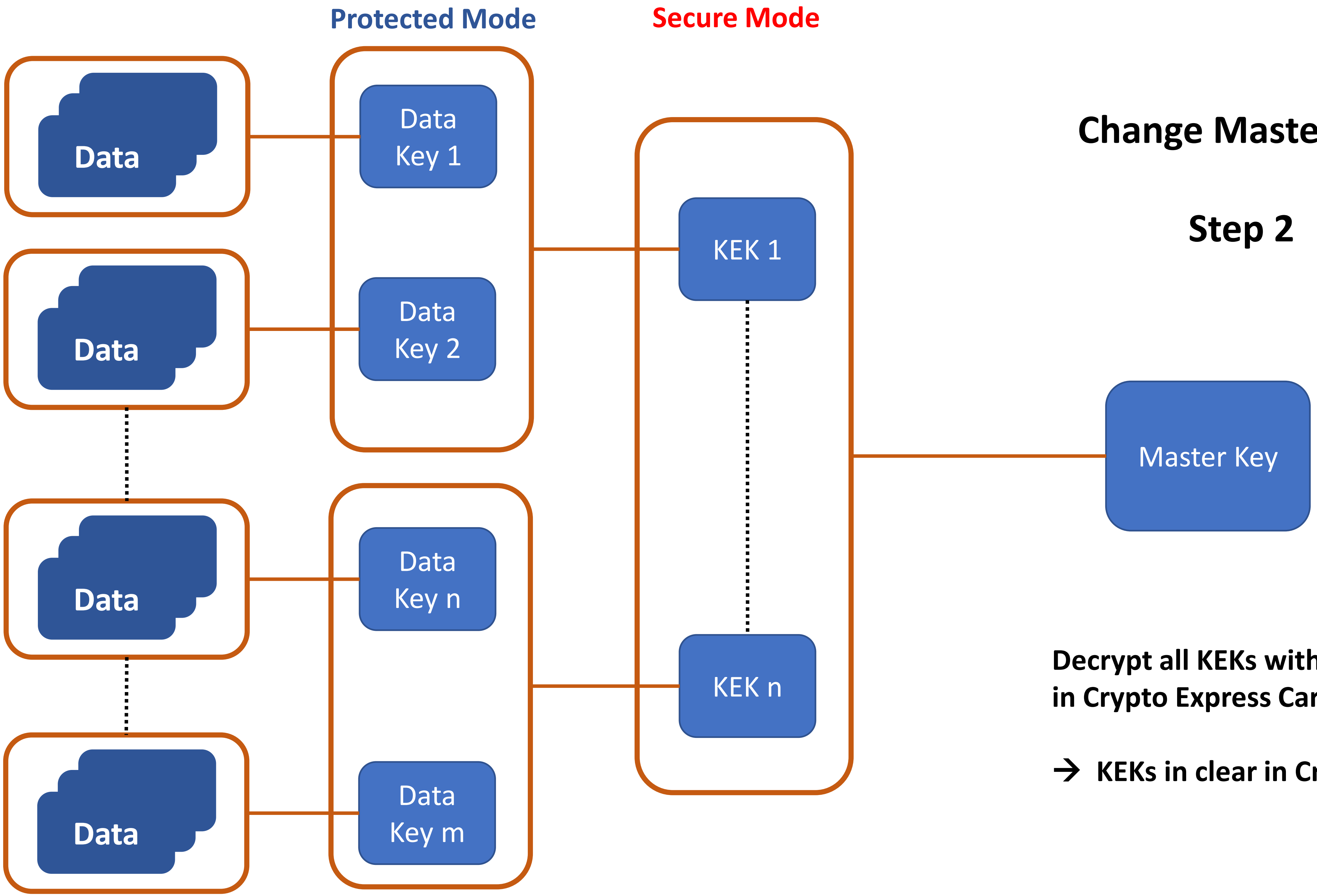
Change Master Key

Step 1

Move all KEKs - Encrypted

from CKDS

into Crypto Express Card

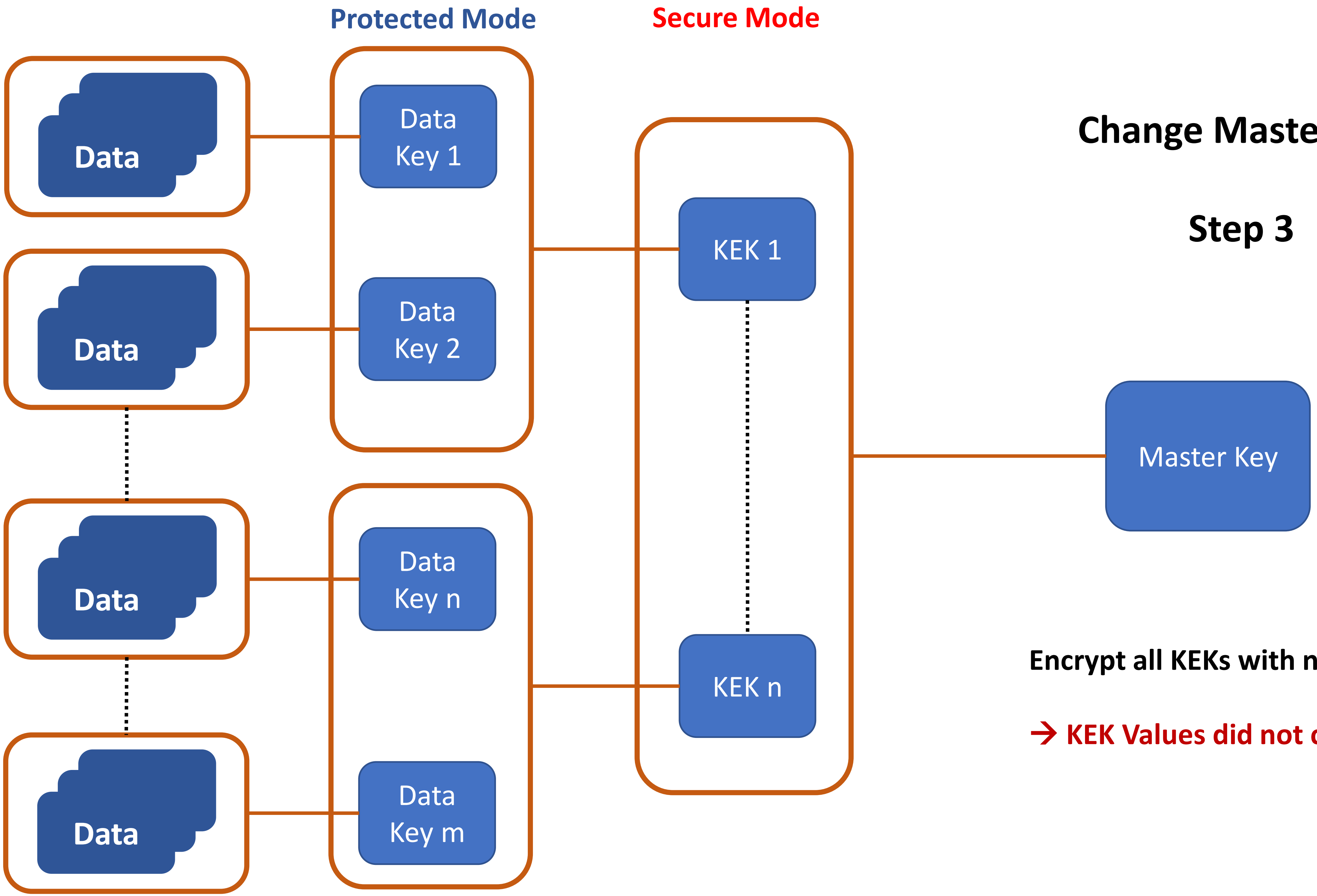


Change Master Key

Step 2

Decrypt all KEKs with old Master Key in Crypto Express Card

→ KEKs in clear in Crypto Express Card



Change Master Key

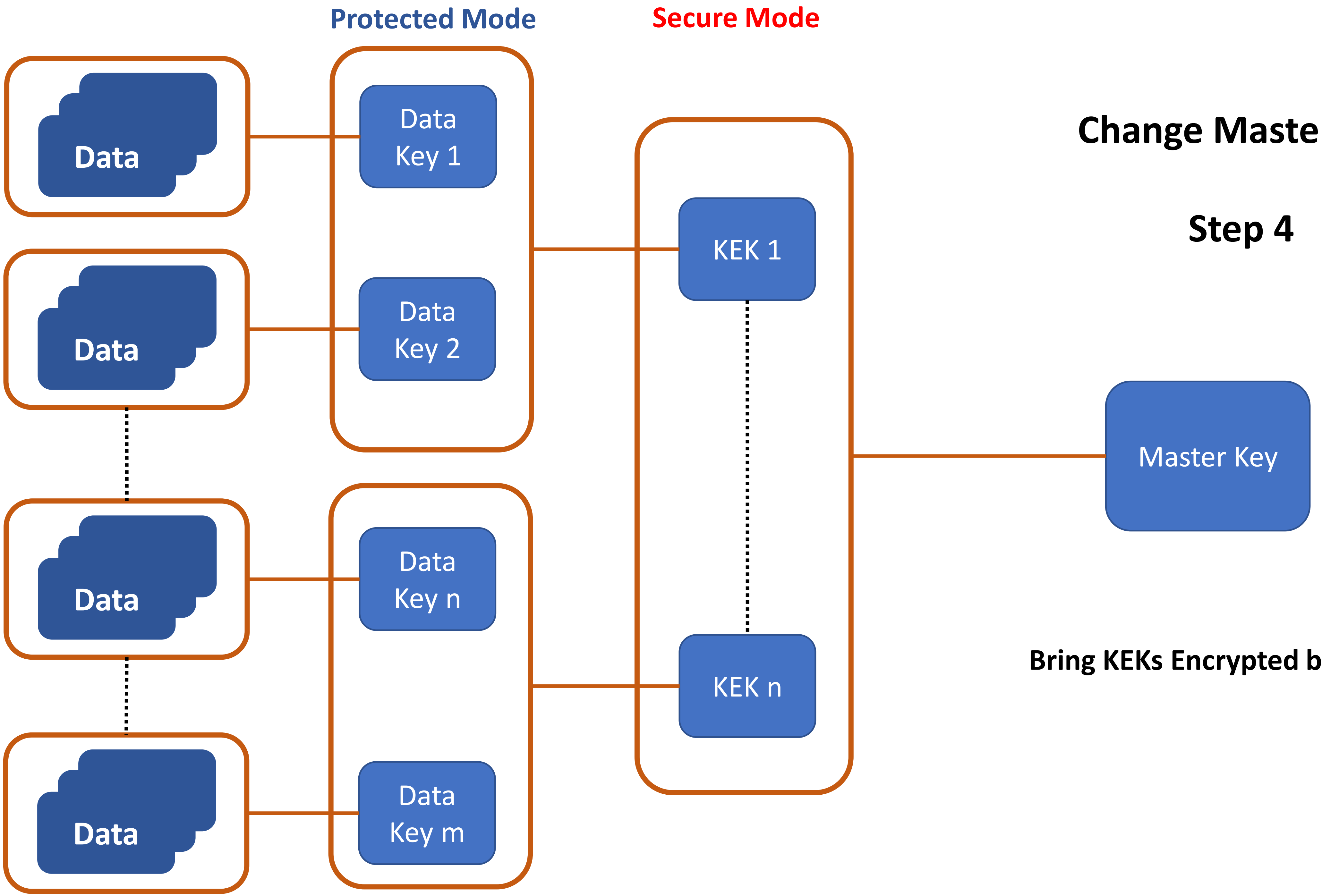
Step 3

Encrypt all KEKs with new Master Key

→ KEK Values did not change

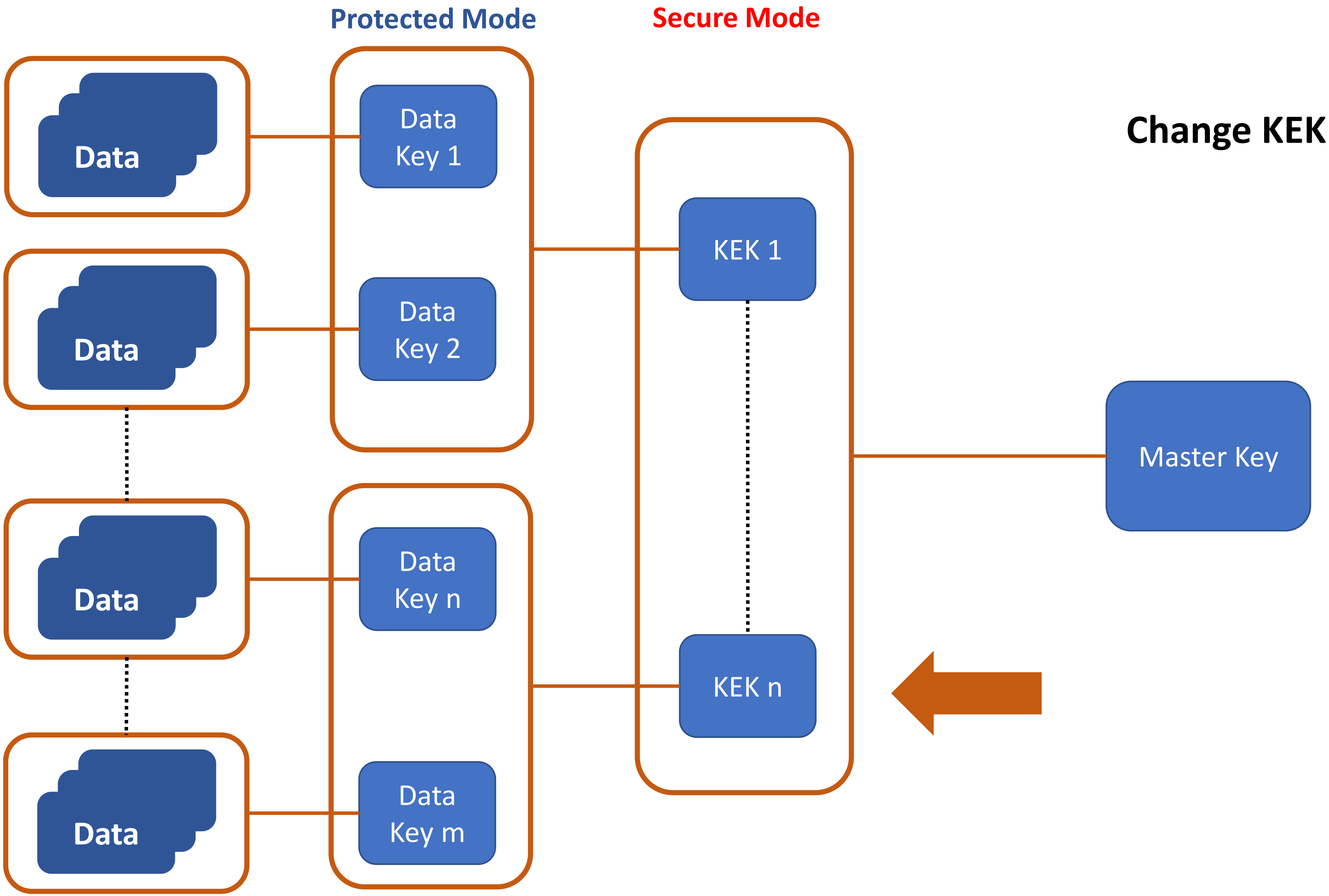
Change Master Key

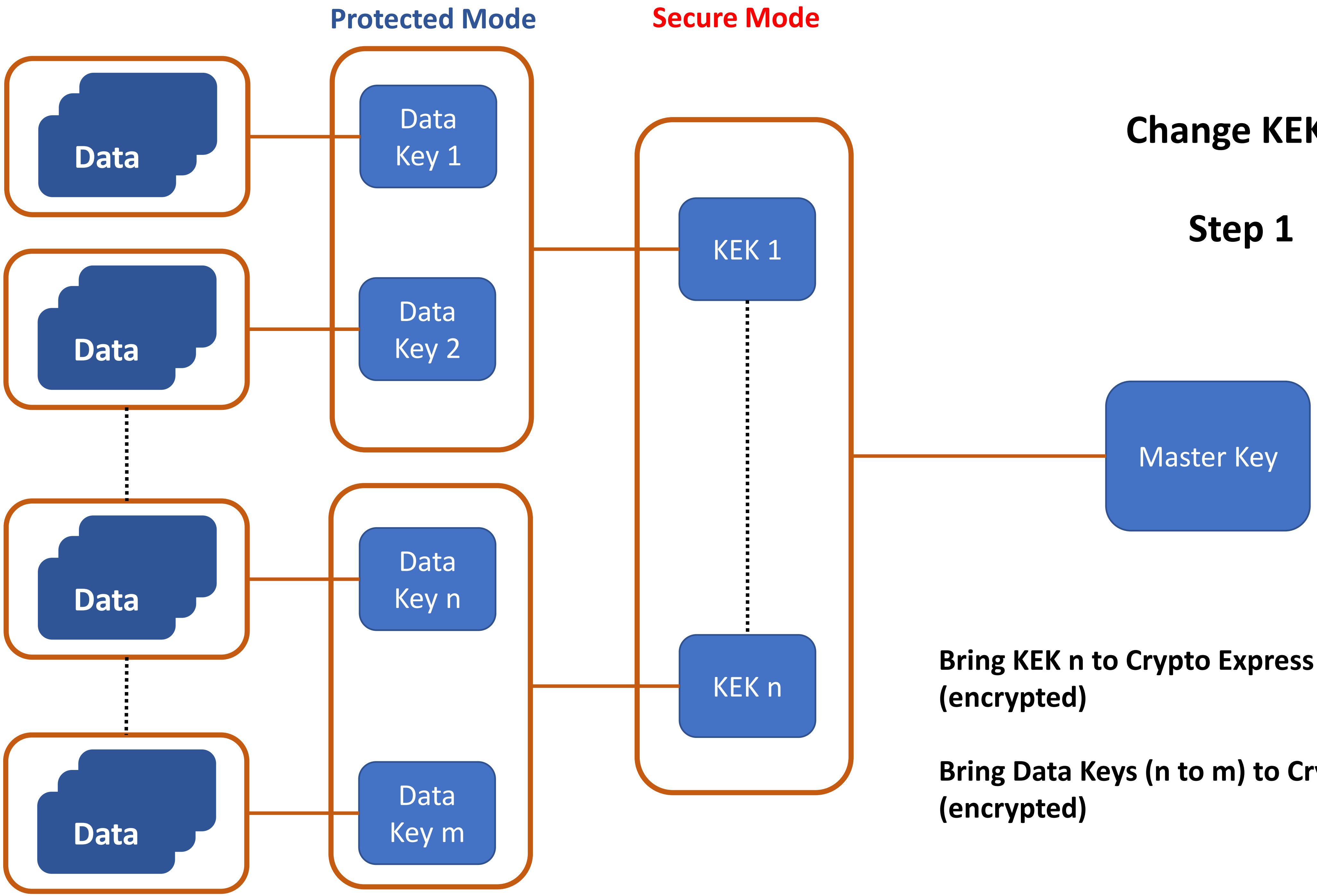
Step 4



Bring KEKs Encrypted back to CKDS

Change KEK n


Protected Mode
Secure Mode
Data
Data
Data
Data
Data Key 1
Data Key 2
Data Key n
Data Key m
KEK 1
KEK n
Master Key



Change KEK n

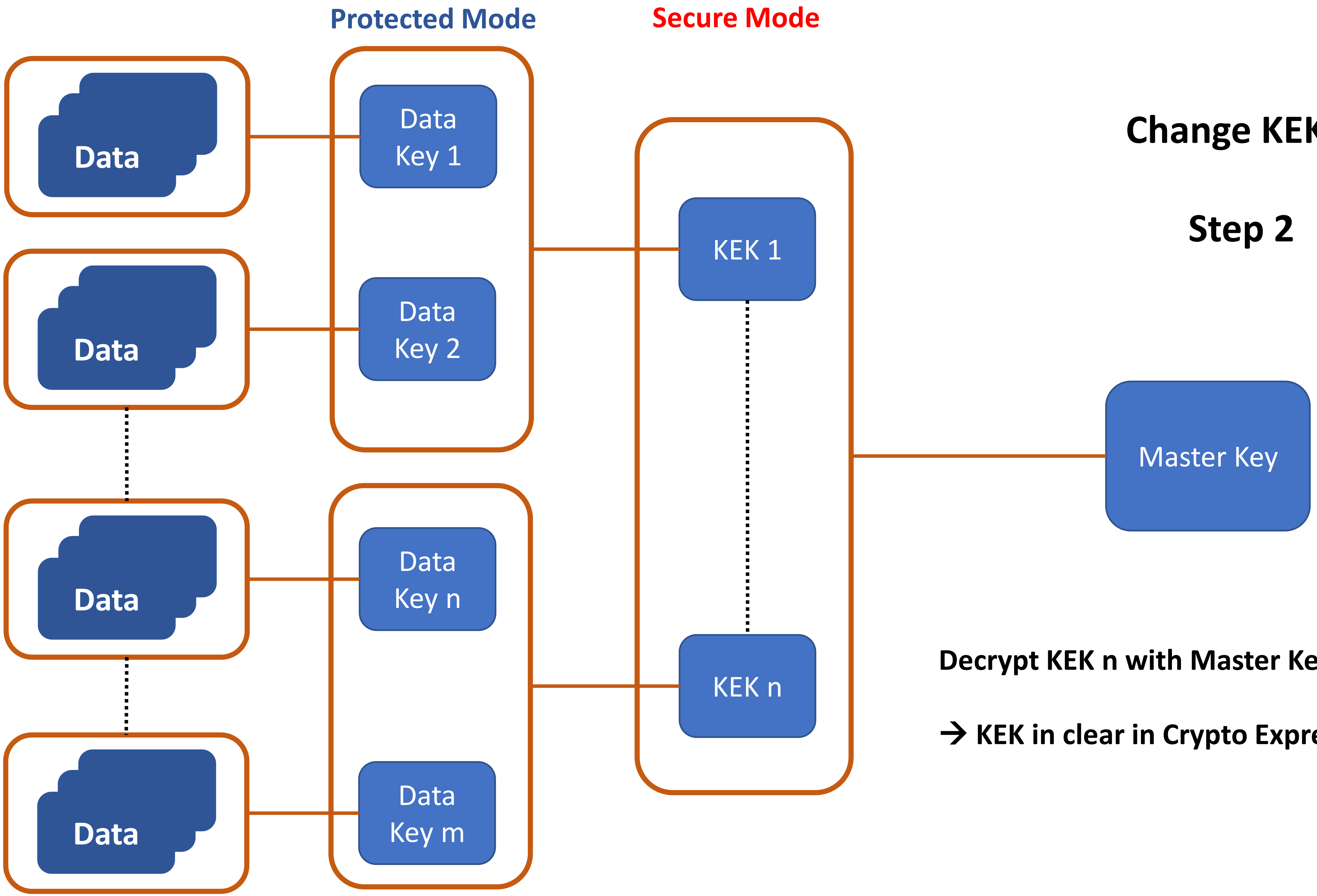
Step 1

Bring KEK n to Crypto Express Card (encrypted)

Bring Data Keys (n to m) to Crypto Express Card (encrypted)

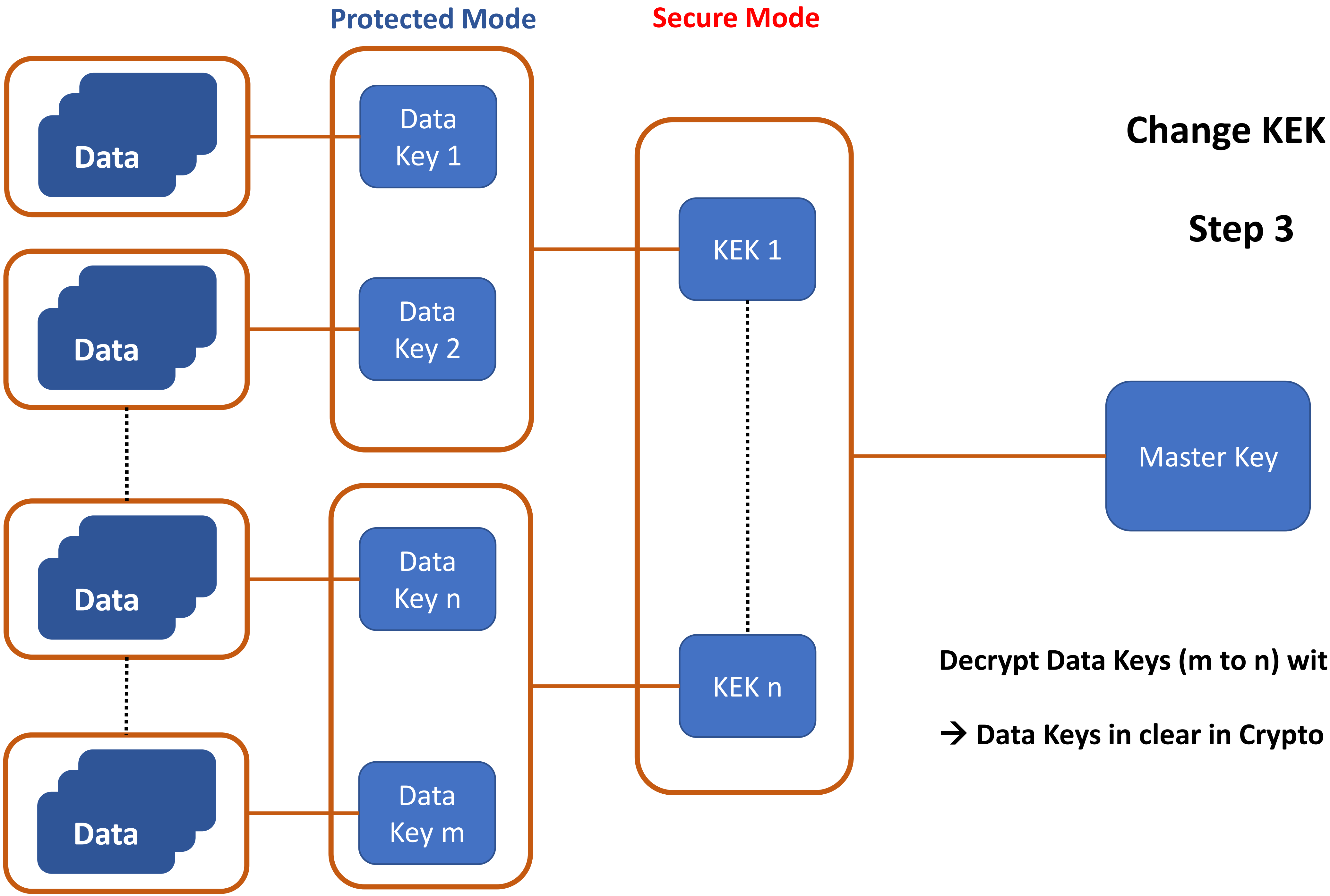
Change KEK n

Step 2



Decrypt KEK n with Master Key

→ KEK in clear in Crypto Express Card



Change KEK n

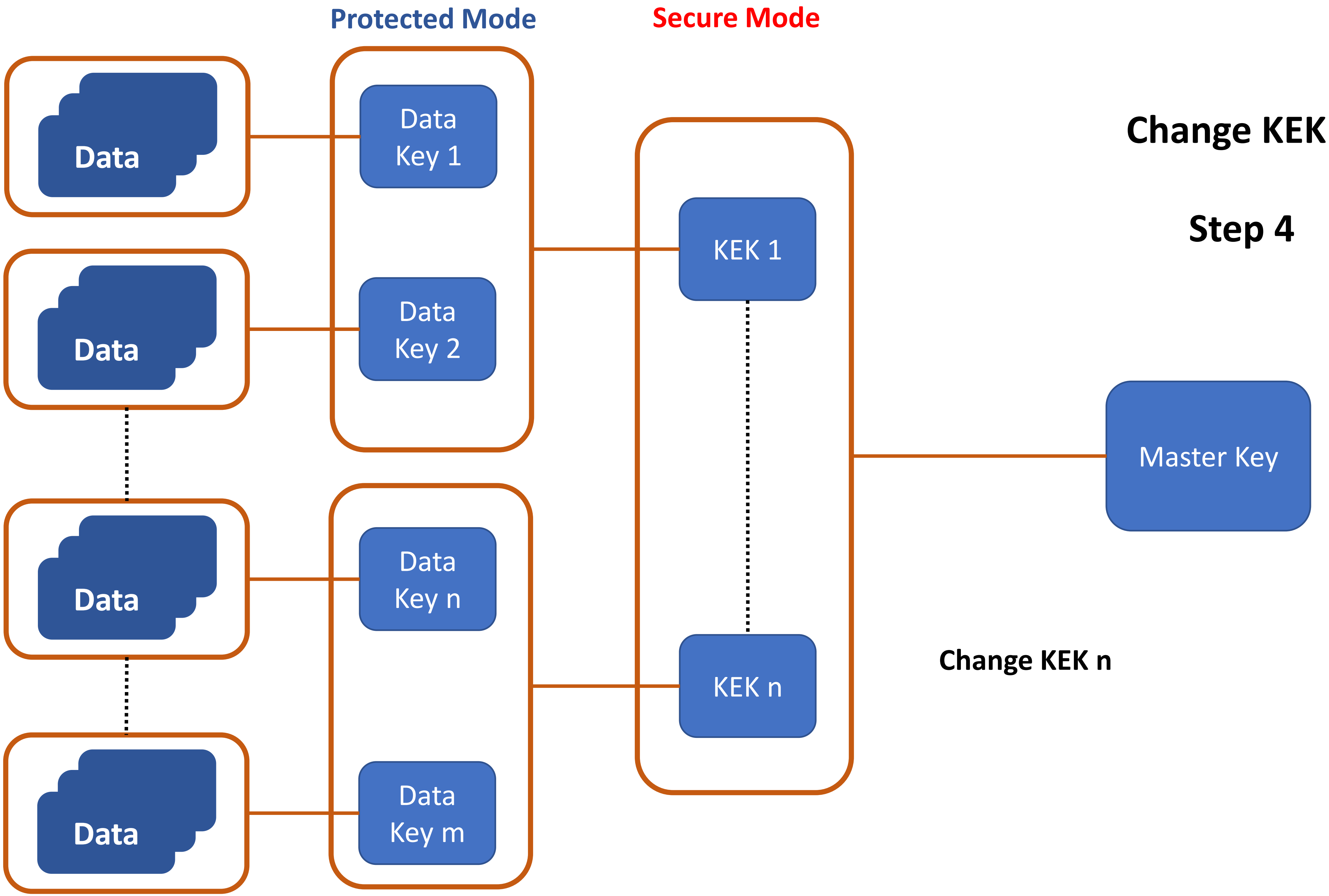
Step 3

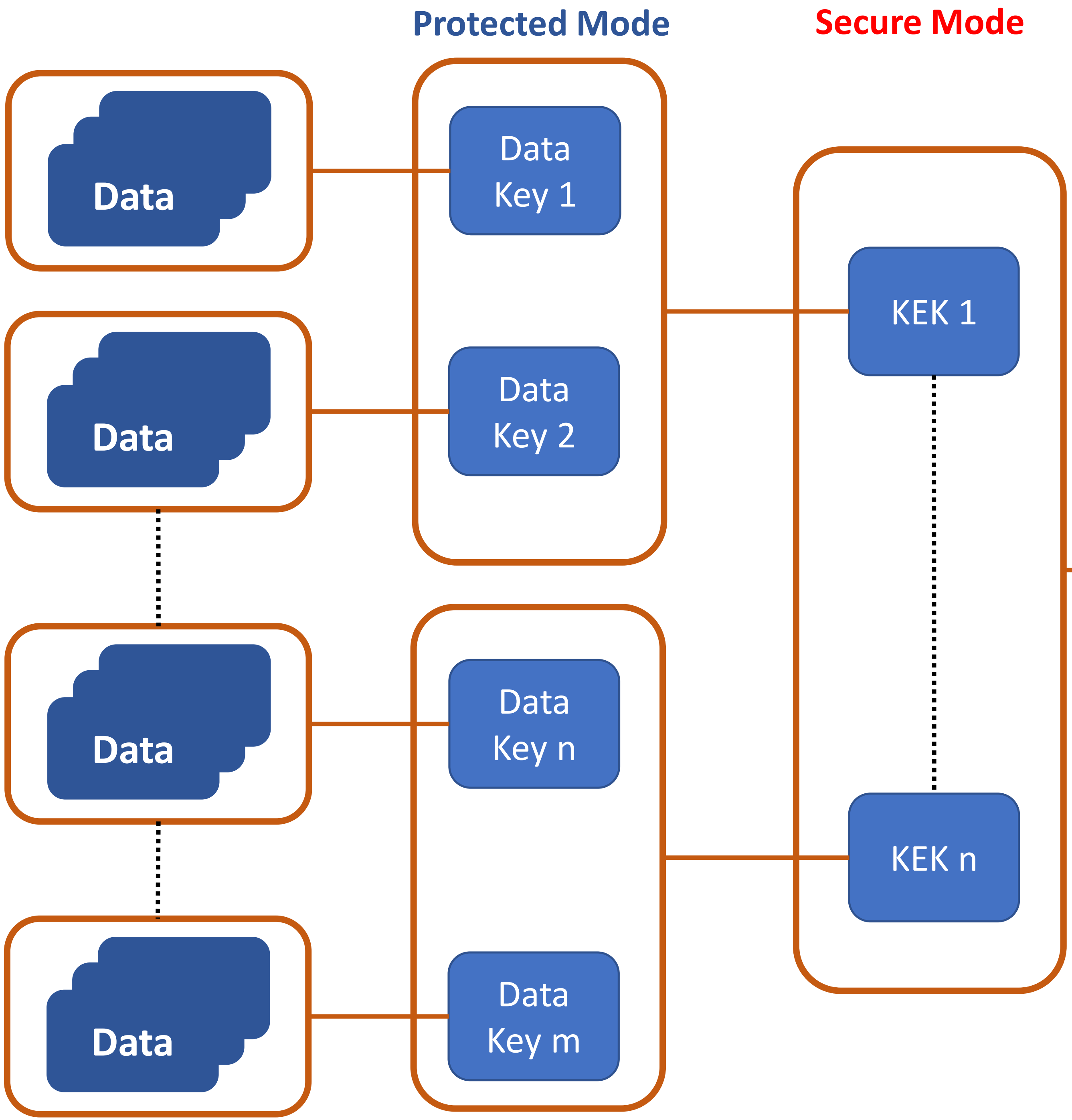
Decrypt Data Keys (m to n) with KEK n

→ Data Keys in clear in Crypto Express Card

Change KEK n

Step 4





Change KEK n

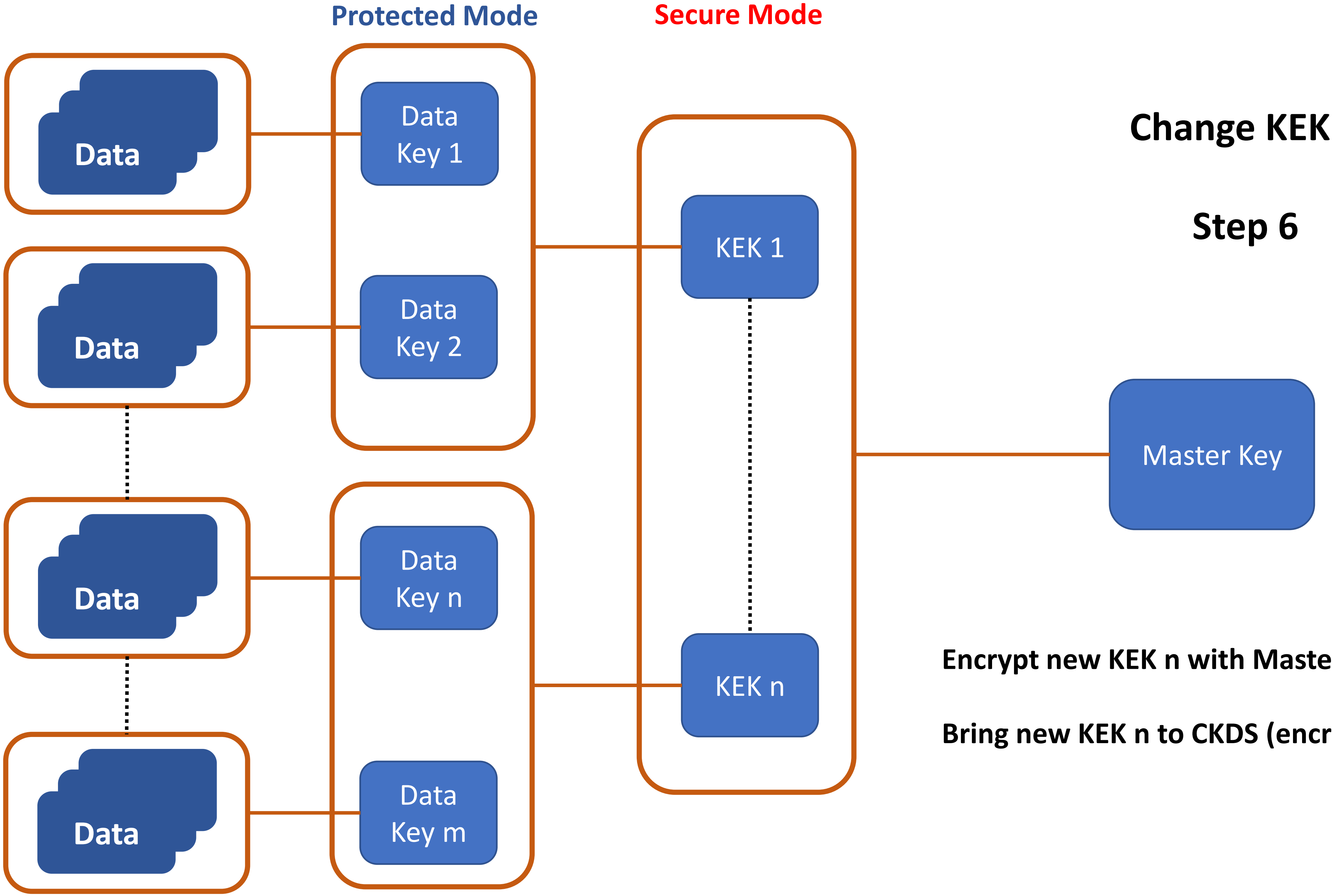
Step 5

Encrypt Data Keys (n to m) with new KEK n

→ Data Keys encrypted in Crypto Express Card

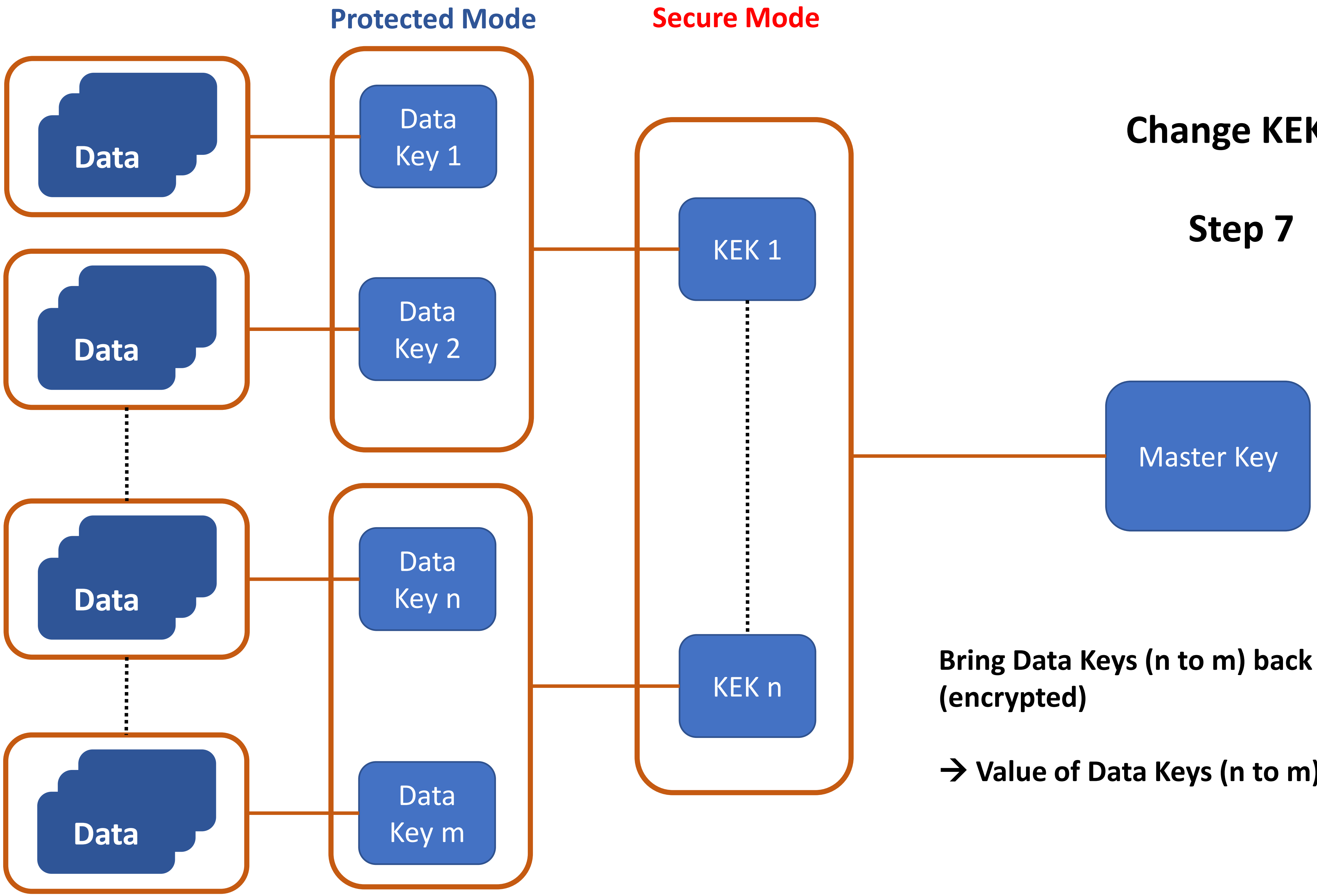
Change KEK n

Step 6



Encrypt new KEK n with Master Key

Bring new KEK n to CKDS (encrypted)



Change KEK n

Step 7

Bring Data Keys (n to m) back to CKDS (encrypted)

→ Value of Data Keys (n to m) did not change

Data Set Level Encryption

Key Management

Other Aspects

Change Data Key (if necessary)

- Archive old key (as an alternative to delete the key)

Data Management (Copy/Dump/Restore)

- Data remains encrypted
- Data without key is unusable

Compress and Encrypt

- Compress first – Encrypt second
- Decrypt first – Decompress second
- Future of zEDC card? but ....

Overhead of Data Set Compression

- z13: approx. 12 - 19%
- z14: approx. 3 - 4%

Master Key Management with TKE recommended

Manage Operational Keys

- f.e. EKMF: Enterprise Key Managament Foundation

Redbook: SG24-8410-00: Getting Started with z/OS Data Set Encryption (June 2018)

Questions