

IBM Cryptographic Subsystem

In 1977, IBM announced a family of products under the name IBM Cryptographic Subsystem, for cryptographic support of communication between terminals and mainframes [IBM77a, IBM77b]. A description of the design principles is given in [EHRS78], [LENN78], and [MATY78]. An independent and similar proposal was made in [EVER78].

The IBM Cryptographic Subsystem is based on the concept of a session key, KS , a key shared by host and terminal and used only for the duration of a session. The basic features of the subsystem are:

1. Two classes of keys, **operational keys** (**session keys** and **device keys**), which perform encryption (encipherment) and **system keys**, are used to encipher operational keys that are stored in the host.
2. No operational key is stored in plaintext in the host.
3. A single **host master key**, KMH , is stored at the host in a secure device capable of executing a small set of instructions. Two variants of the master key, $KMH0$ and $KMH1$, are obtained by complementing certain bits in KMH .
4. **Device (or terminal) master keys**, $KMT_1, KMT_2, \dots, KMT_N$, are stored in a secure manner at the devices (terminals). A host table contains $(DES[KMH_1, KMT_i])$ for $i = 1, 2, \dots, N$.
5. A session key KS is an operational key used to encipher transmissions between the host and a device (terminal). The session key is generated by the host at the initiation of the session.
6. There are four nonprivileged instructions for enciphering/deciphering data: $ECPH$, $DCPH$, DMK , and $RFMK$.

Figure 1 shows the operation of the $ECPH$ (encipher data) instruction, which has the form:

$$ECPH(DES^{-1}[KMH0, KS], PLAIN) = DES[KS, PLAIN] = CIPHER$$

The session key is stored at the host encrypted using KMH0. When a block of plaintext is to be transmitted to a terminal, the session key must first be decrypted, and then used to encrypt the plaintext.

Figure 2 shows the operation of the DCPH (decipher data) instruction, which has the form:

$$\text{DCPH}(\text{DES}^{-1}[\text{KMH0}, \text{KS}], \text{DES}[\text{KS}, \text{PLAIN}]) = \text{PLAIN} = \text{DES}^{-1}[\text{KS}, \text{CIPHER}]$$

In this case, the host receives encrypted plaintext, recovers the session key using KMH0, and uses the session key to recover the plaintext.

Figure 3 shows the operation of DMK (decipher under terminal key), which is an instruction executed at a terminal (device), which has the form:

$$\text{DMK}(\text{KMT}_i, \text{DES}[\text{KMT}_i, \text{KS}]) = \text{DES}^{-1}[\text{KMT}_i, \text{DES}[\text{KMT}_i, \text{KS}]] = \text{KS}$$

Figure 4 shows a translation facility, RFMK (reencipher from master key), which has the form:

$$\text{RFMK}(\text{DES}[\text{KMH1}, \text{KMT}_i], \text{DES}[\text{KMH0}, \text{KS}]) = \text{DES}[\text{KMT}_i, \text{KS}]$$

A session linking the i th device (terminal) to the host processor involves the following steps:

1. A pseudorandom number RN is generated at the host by repeated DES encryption of the time-of-day clock. RN is interpreted as the encryption of the session key KS under KMH0.

$$\text{RN} = \text{DEX}[\text{KMH0}, \text{KS}]$$

RN is stored at the host in a table associated with the i th device for the duration of the session.

2. The session key must be made available to the i th device. The translation facility is used for this purpose; RFMK with input arguments (1) the encryption of KMT_i under $KMH1$, and (2) RN, which yields

$$DES[KMT, KS] = RFMK(DES[KMH1, KMT_i], DES[KMH0, KS])$$

This is transmitted by the host to the i th device.

3. The i th device, having KMT_i , uses DMK to obtain

$$KS = DMK(KMT_i, DES[KMT_i, KS]) = DES^{-1}[KMT_i, DES[KMT_i, KS]]$$

At the end of step 3, both parties to the communication, the host and the terminal, have now established and exchanged a common key KS. Plaintext can be encrypted with the key KS at the host with ECPH, and plaintext can be recovered at the host with DCPH.

References

- EHR78** Ehrtam, W., et al. "A Cryptographic Key Management Scheme for Implementing the Data Encryption Standard." *IBM Systems Journal*, Vol 17, No 2, 1978.
- EVER78** Everton, J. "A Hierarchical Basis for Encryption Key Management in a Computer Communication Network." Proceedings ICC'78, 1978.
- IBM77a** IBM. *Programmed Cryptographic Facility Program Product - General Information Manual*. IBM Systems Library, GC28-0941, 1977.
- IBM77b** IBM. *IBM 3848 Cryptographic Unit Product Description and Operating Procedures*. IBM Systems Library, GC22-7073, 1977.
- LENN78** Lennon, R. "Cryptographic Architecture for Information Security." *IBM Systems Journal*, Vol 17, No 2, 1978.
- MATY78** Matyas, S., and Meyer, C. "Generating, Distribution, and Installation of Cryptographic Keys." *IBM Systems Journal*, Vol 17, No 2, 1978.

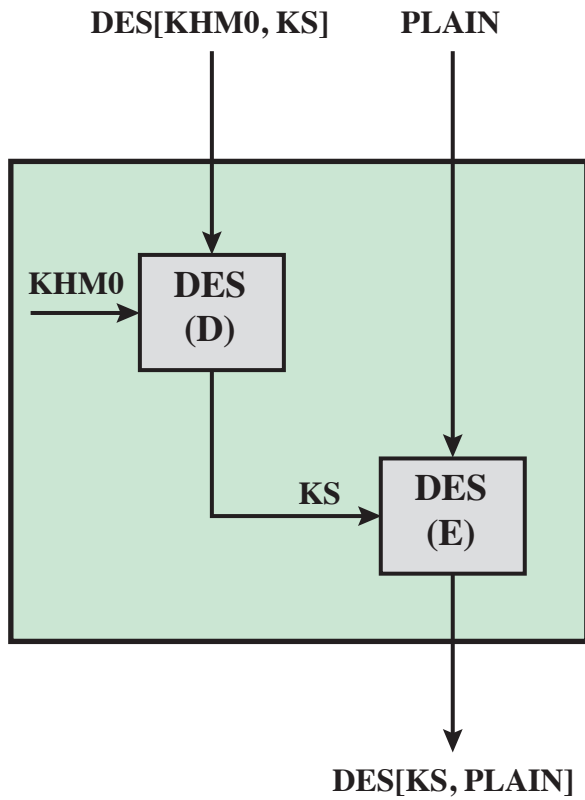


Figure 1 ECPH: encipher data

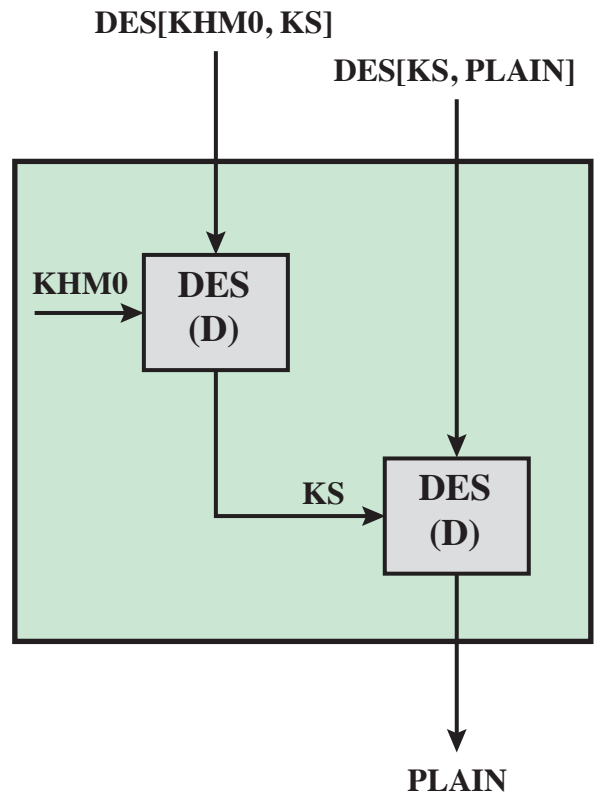


Figure 2 DCPH: decipher data

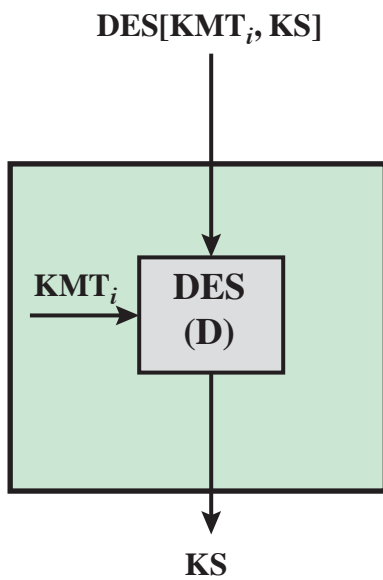


Figure 3 DMK: decipher under terminal (device) key

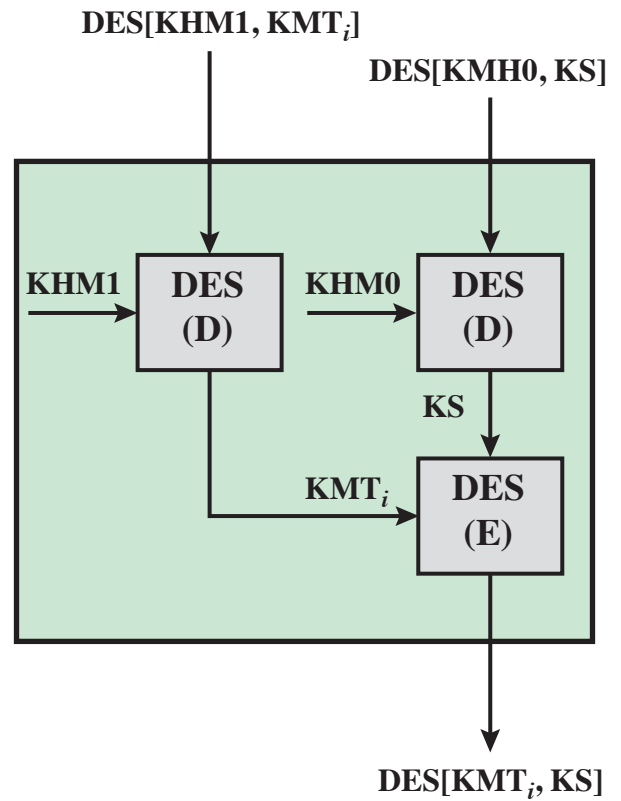


Figure 4 RFMK: recipher from master key