



Figure 45: Encryption

If the following occurs, the **ERROR** event is generated, the CCM stops, and the **ERRORSTATUS** register will report the type of error that triggered the **ERROR** event:

- The **IN.PTR** job list ends before reading out the complete CCM data structure
- The **OUT.PTR** job list ends before writing out the complete encrypted CCM data structure
- The CCM is not able to operate fast enough to run concurrently with the RADIO as the RADIO transmits the encrypted packet.
- The EasyDMA engine encounters an error, see [EasyDMA and ERROR event](#) on page 239

Any values of $l(m)$ and $l(a)$ are allowed. If encrypting empty packets, i.e. $l(m) = l(a) = 0$, no encryption will take place; the **END** event is generated, and CCM operation is stopped.

For Bluetooth (**MODE.PROTOCOL=BLE**), valid packets with 0 payload ($l(a)$ is larger than 0 but $l(m)$ is 0) will not be authenticated but instead moved unmodified through the AES CCM peripheral, and thus no MAC will be generated.

For IEEE 802.15.4 (**MODE.PROTOCOL=IEEE802154**), valid packets with 0 payload ($l(a)$ is larger than 0 but $l(m)$ is 0) will be authenticated, and thus a MAC will be generated as part of the output data.

8.4.2.2 Decryption

During packet decryption, CCM will read the encrypted packet located in memory at the address specified in the **IN.PTR** pointer, decrypt the packet, authenticate the packet's MAC field and generate the appropriate MAC status.

The encrypted message in (c) , is decrypted and authenticated together with additional authenticated data (a) and then matched against the decrypted MAC value. The decrypted MAC value is part of (c) . Bits in the first byte of the data can be masked away before calculating the MAC value by configuring the **ADATAMASK** register. This is useful for Bluetooth header masking. For protocols other than Bluetooth, the **ADATAMASK** register must be set to 0xFF for correct CCM operation; the reset value is configured to support Bluetooth.