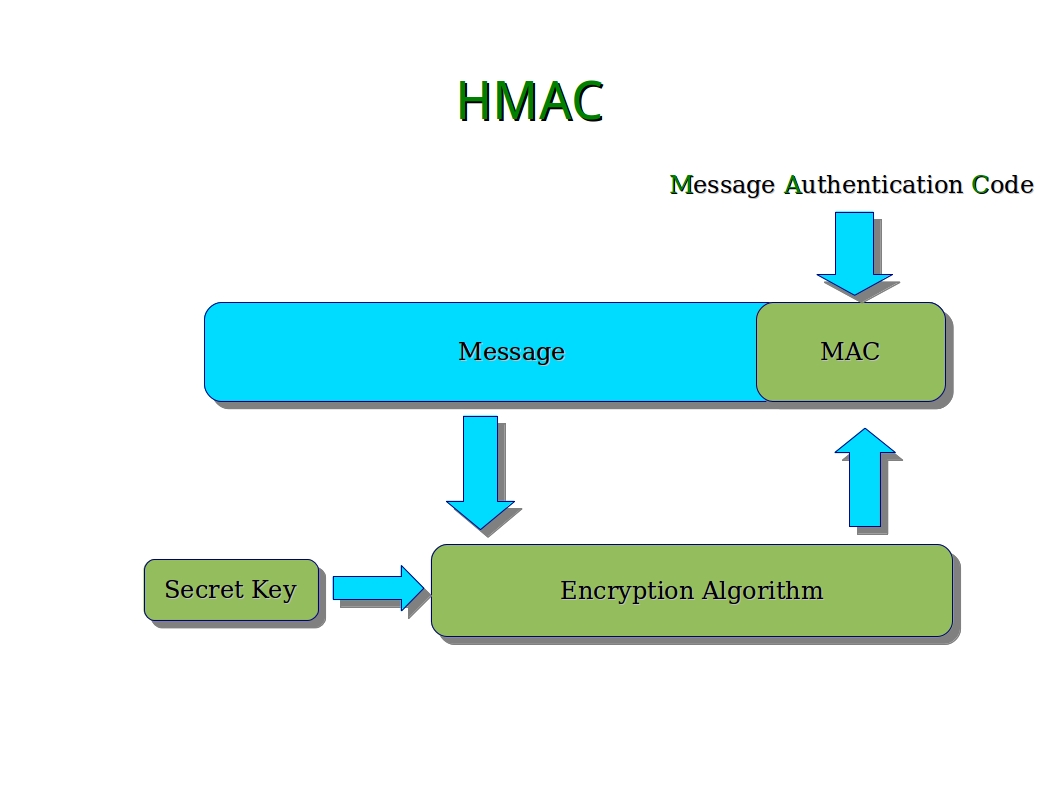
**1. Integrity Protection using HMAC and AES**

**Real-Life Example: Bank Transactions**

* **Scenario**: Imagine you're transferring money from your bank account to someone else’s. You and the bank exchange transaction data over the internet.
* **Problem**: During transmission, an attacker might modify the transaction data (e.g., changing the amount to a larger number). This is called a **man-in-the-middle attack**.
* **How the code helps**:
  + When the transaction details (such as the amount and account number) are transmitted, the bank and you both calculate a **MAC (Message Authentication Code)** based on the transaction details.
  + The **MAC** is like a digital fingerprint of the data that can’t be changed without altering the data itself.
  + If someone intercepts and modifies the data, the **MAC** will no longer match when you (the receiver) check it, and you’ll know that the transaction has been tampered with.
  + This ensures **data integrity**—the transaction data remains exactly the same as it was sent.

**Why it's Used:**

* **In banking** and other financial services, you want to make sure that the details of a transaction are **secure** and **not tampered with** before being processed.



**2. AES Encryption and Decryption in CBC Mode**

**Real-Life Example: Secure Messaging Apps (e.g., WhatsApp)**

* **Scenario**: When you send a private message through a messaging app like WhatsApp, you want to ensure that no one other than the recipient can read it, even if someone intercepts the message in transit.
* **Problem**: Without encryption, if someone hacks into the communication channel, they can read all the messages being sent.
* **How the code helps**:
  + **AES encryption** in **CBC mode** ensures that your message is **encrypted** into unreadable ciphertext before it leaves your phone.
  + The message is **padded** (if needed) to ensure it fits the encryption block size.
  + An **IV (Initialization Vector)** is used to ensure that the same message, when encrypted multiple times, results in different ciphertexts each time, making it more secure.
  + Once the message reaches the recipient, it is decrypted using the same key, and the original message is revealed.
  + Even if someone intercepts the message, they won’t be able to read it because it’s encrypted. Only the recipient with the correct decryption key can read it.

**Why it's Used:**

* **In messaging apps**, encryption protects **confidentiality** by ensuring that only the sender and the recipient can read the messages.
* **CBC mode** ensures that each message is securely encrypted, and the IV ensures the encryption is unpredictable each time, even for identical messages.

**Key Takeaways:**

1. **Integrity Protection (Code 1)**: In scenarios like **bank transactions** or **secure file transfers**, integrity protection ensures that the data you send (such as your transaction details) is not altered or tampered with during transmission.
2. **AES Encryption (Code 2)**: In **messaging apps** or **secure data storage**, AES encryption keeps your data safe from unauthorized access by making it unreadable to anyone except the intended recipient.

Both these techniques are essential for **security** and **confidentiality** in our everyday digital interactions.

