A Brief Discussion on AES Modes of Operation

CRS2404

September 1, 2025

1 The Insecurity of Electronic Codebook (ECB) Mode

When first approaching block ciphers like AES, the most straightforward way to encrypt a message longer than a single block is what's known as the Electronic Codebook (ECB) mode. The process is simple: you break the plaintext message into fixed-size blocks (16 bytes for AES) and encrypt each block independently using the same secret key. While simple, this method contains a critical, and often fatal, security flaw.

The fundamental problem with ECB lies in its determinism. Since each block is encrypted in exactly the same way, **any two identical plaintext blocks will always produce the exact same ciphertext blocks**. This means that patterns, structure, and repetition in the original data are preserved and remain visible in the encrypted output.

To see why this is so bad, consider the classic example of encrypting an image file. An unencrypted image of a penguin, for instance, has large areas of uniform color (black, white, yellow). If we encrypt this image using ECB mode, all the blocks corresponding to the white background will be transformed into the same encrypted block. All the blocks for the black parts of the penguin will be transformed into another, different, encrypted block. The result is that the overall shape and features of the penguin remain clearly visible in the ciphertext. An attacker might not know the exact colors, but they can certainly still see the penguin, leaking a massive amount of information.

This pattern preservation makes ECB mode completely unsuitable for encrypting structured data of any kind, not just images. Text, executable files, and structured data formats all contain repeating patterns that would be exposed by ECB. Furthermore, this mode is vulnerable to other attacks, such as:

- **Replay Attacks:** An attacker can copy and paste blocks from one message into another. For example, if a block corresponding to "$1,000,000" is identified, an attacker could splice it into a different transaction.
- **Block Rearrangement:** An attacker can reorder, duplicate, or delete blocks to manipulate the final decrypted message without ever needing to know the key.

For these reasons, ECB mode is considered insecure for almost all general-purpose cryptographic applications and should be avoided.

2 Recommended Alternatives to ECB

To fix the flaws of ECB, more advanced modes of operation were developed. These modes ensure that even if you encrypt the same block multiple times, it will result in different ciphertext each time. This property is known as semantic security.

For modern applications, my strong recommendation would be to use **GCM (Galois/Counter Mode)**. GCM is a type of *Authenticated Encryption with Associated Data* (AEAD) scheme, which has become the gold standard for modern cryptography. It provides three essential security guarantees in one efficient package:

1. **Confidentiality:** It encrypts the message so no one can read it, just like any other mode. It achieves this by using Counter (CTR) mode internally, which acts like a stream cipher.

2. **Integrity:** It guarantees that the message has not been tampered with or altered in any way. If a single bit of the ciphertext is flipped, it will be detected upon decryption.

3. **Authenticity:** It cryptographically verifies the source of the message, ensuring it came from someone who possesses the secret key.

The power of GCM is that it bundles these features together. Older modes like **CBC (Cipher Block Chaining)**, while a massive improvement over ECB, only provide confidentiality. In CBC, each plaintext block is XORed with the *previous* ciphertext block before encryption, creating a dependency chain that hides patterns. However, you would still need to combine it with a separate Message Authentication Code (like HMAC) to get integrity and authenticity. Getting this combination right is notoriously difficult and has led to many real-world vulnerabilities.

GCM is also highly performant. Because it uses CTR mode as its foundation, the encryption/decryption of each block can be done in parallel, making it very fast on modern hardware.

In summary, while older modes like CBC were once standard, they are now considered legacy. For any new system or protocol being built today, an AEAD mode should be the default choice, and **GCM is the most widely used, well-vetted, and performant option available**. It provides a complete security solution that protects against a wide range of attacks that confidentiality-only modes leave open.