# Block Cipher Modes of Operation

Luke Anderson

luke@lukeanderson.com.au

23rd March 2018

University Of Sydney

# Overview


## 1. Crypto-Bulletin

## 2. Modes Of Operation

2.1 Evaluating Modes

2.2 Electronic Code Book (ECB)

2.3 Cipher Block Chaining (CBC)

2.4 Output Feedback Mode (OFB)

2.5 Counter Mode (CTR)

2.6 Galois/Counter Mode (GCM)

# Crypto-Bulletin

# Crypto-Bulletin

## 15-Year-old Finds Flaw in Ledger Crypto Wallet

https://krebsonsecurity.com/2018/03/15-year-old-finds-flaw-in-ledger-crypto-wallet/

## GrayKey iPhone unlocker poses serious security concerns

https://blog.malwarebytes.com/security-world/2018/03/graykey-iphone-unlocker-poses-serious-security-concerns/

## Github: Weak cryptographic standards removed

https://blog.github.com/2018-02-23-weak-cryptographic-standards-removed/

# Modes Of Operation

# Cipher Modes of Operation

Once a key $k$ is chosen and loaded into a block cipher, $E_k$ only operates on single blocks of data.

1. Block size usually small (16 byte blocks for AES)
2. Message to be sent usually large (web page + assets $\approx$ 500kB)
3. Need a way to repeatedly apply the cipher with the same key to a large message.

By using different *modes of operation*, messages of an arbitrary length can be split into blocks and encrypted using a block cipher.

Each mode of operation describes how a block cipher is repeatedly applied to encrypt a message and each has certain advantages and disadvantages.

# Evaluating Block Ciphers & Modes

To evaluate a cipher and a mode of operation, examine:

**Key Size**:
Upper bound on security, but longer keys add costs (generation, storage, etc.)

**Block Size**:
Larger is better to reduce overheads, but is more costly.

**Estimated Security Level**:
Confidence grows the more it is analysed.

**Throughput**:
How fast can it be encrypted/decrypted?
Can it be pre-computed? Can it be parallelised?

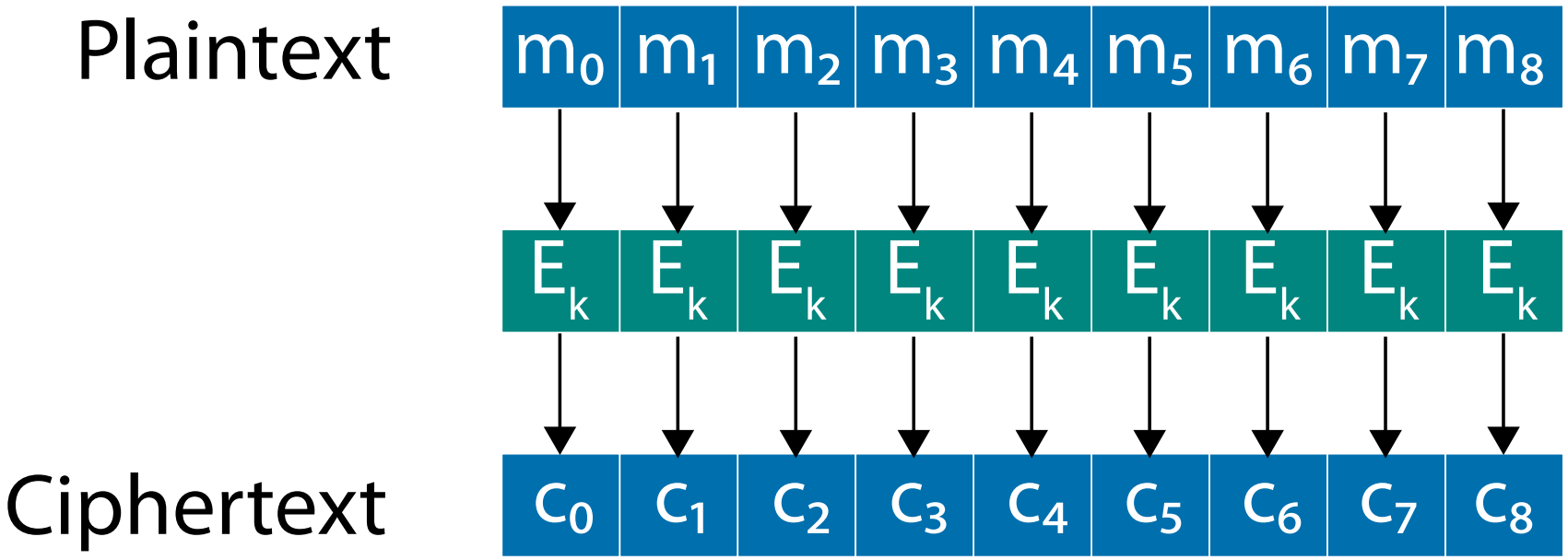
**Error Propagation**:
What happens as a result of bit errors or bit loss?

The first two points above are relevant only to the cipher, while the last three are relevant to both the cipher and a mode of operation.

# Electronic Code Book (ECB)

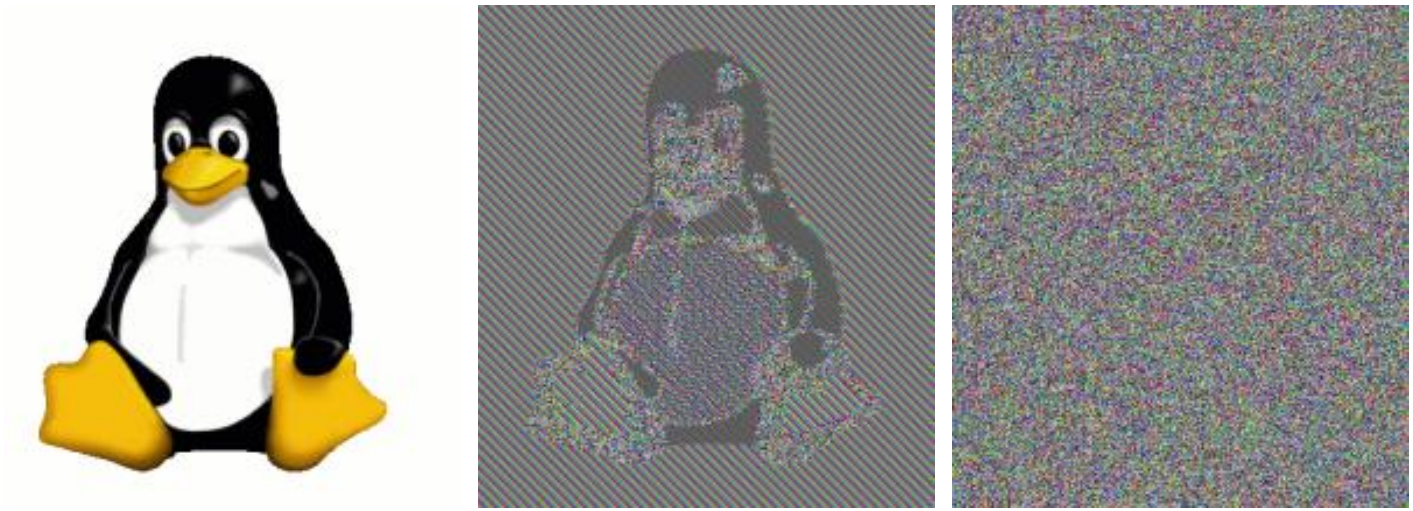
**Electronic Code Book (ECB) encrypts each block separately.**

ECB is generally an insecure and naïve implementation, it is vulnerable to a range of attacks; including dictionary and frequency attacks. *It should never be used.*



# Electronic Code Book (ECB)

The problem with ECB:



(a) Original Image (b) ECB mode (c) Other mode

Encryption of Tux[1] image.

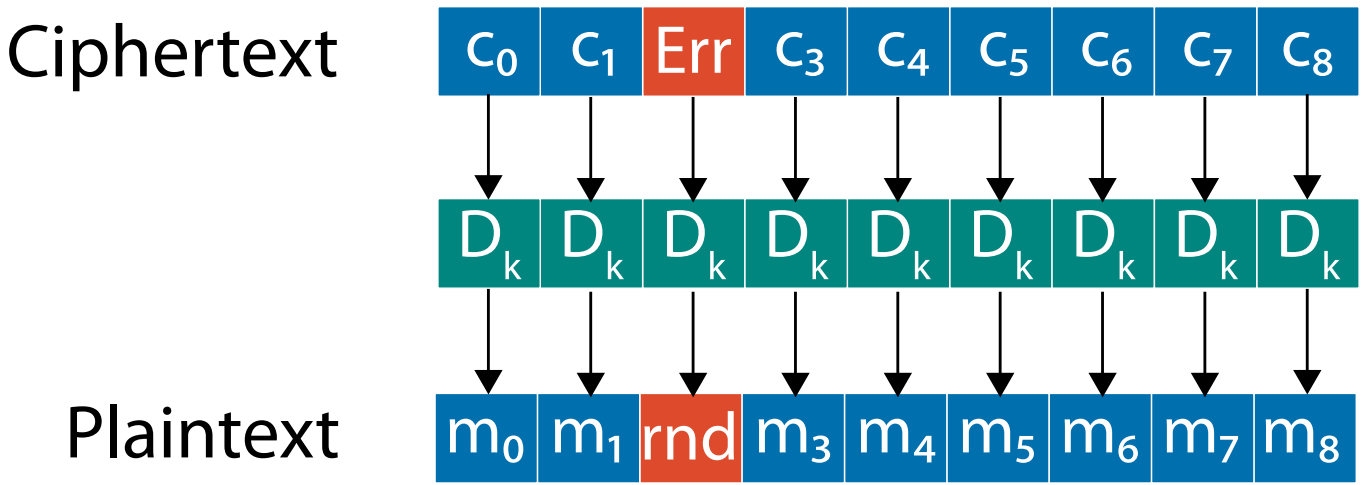
It's a substitution cipher, with blocks instead of letters!

[1] Tux is the Linux mascot

# ECB Properties

**Identical plaintext blocks result in identical ciphertext blocks**
Since blocks are enciphered independently, a reordering of ciphertext blocks results in reordering of plaintext blocks.
ECB is thus not recommended for messages $> 1$ block in length.

**Error propagation**: Bit errors only impact the decoding of the corrupted block (block will result in gibberish)

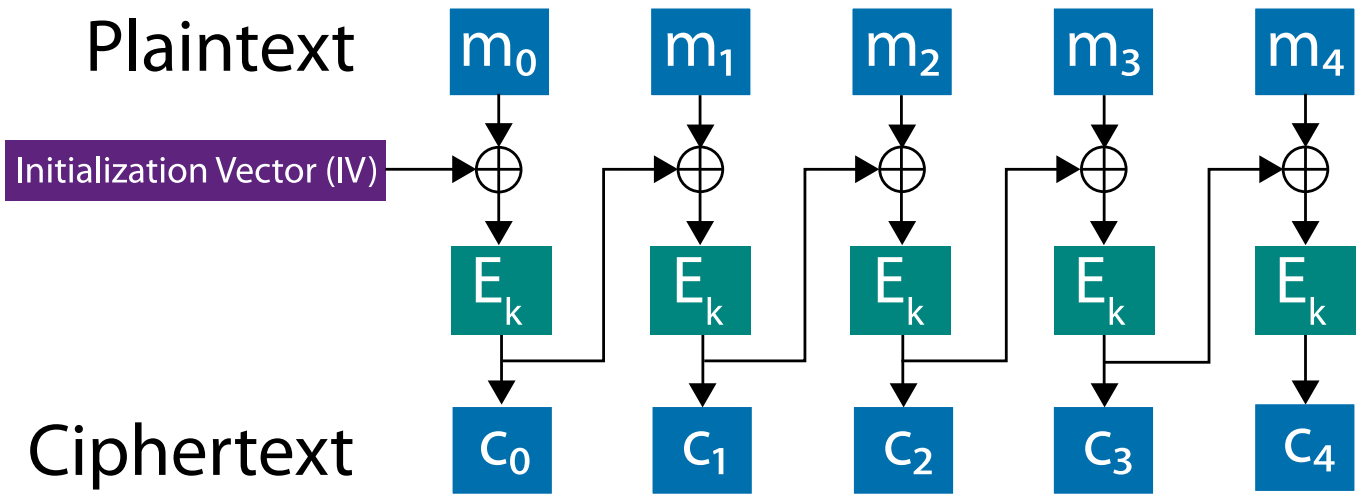



Error propagation in ECB

# Cipher Block Chaining (CBC)

In **Cipher Block Chaining (CBC)** blocks are chained together using XOR.

The **Initialisation Vector (IV)** is a random value that is transmitted in the clear that ensures the same plaintext and key does not produce the same ciphertext.




CBC Mode Encryption

# CBC Properties

Identical plaintexts result in identical ciphertexts when the same plaintext is enciphered using the same key and IV.
Changing at least one of $[k, IV, m_0]$ affects this.

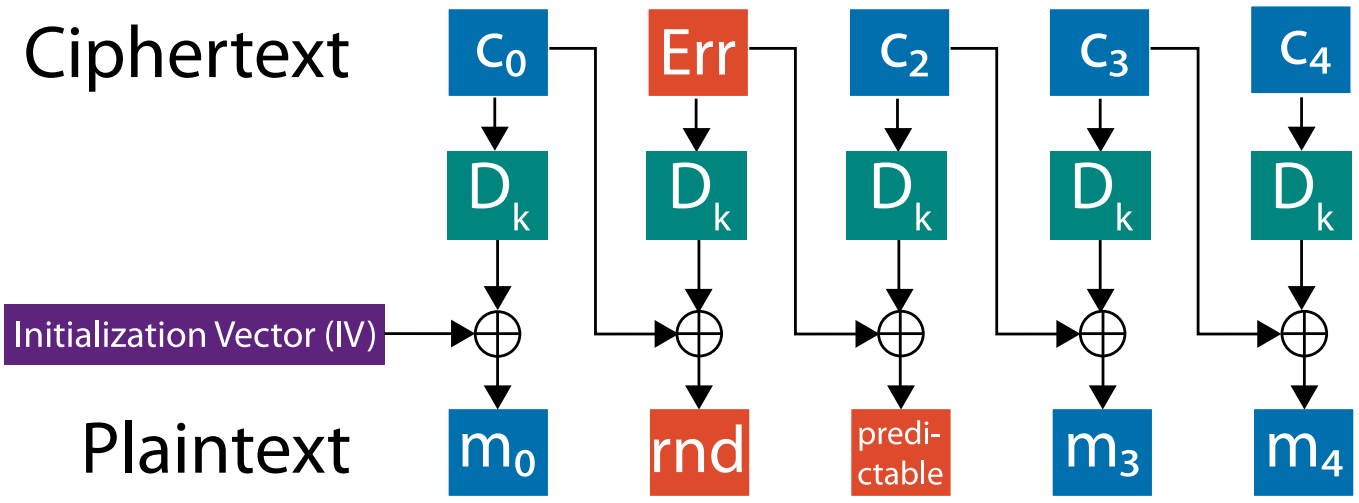
Rearrangement of ciphertext blocks affects decryption, as ciphertext part $c_j$ depends on all of $[m_0, m_1, \cdots, m_j]$.

**Error propagation**:
Bit error in ciphertext $c_j$ affects deciphering of $c_j$ and $c_{j+1}$.Recovered block $m'_j$ typically results in random bits.
Bit errors in recovered block $m'_{j+1}$ are precisely where $c_j$ was in error.
Attacker can cause predictable bit changes in $m_{j+1}$ by altering $c_j$.

**Bit recovery**:
CBC is self-synchronising in that if a bit error occurs in $c_j$ but not $c_{j+1}$, then $c_{j+2}$ correctly decrypts to $m_{j+2}$.

# CBC Decryption

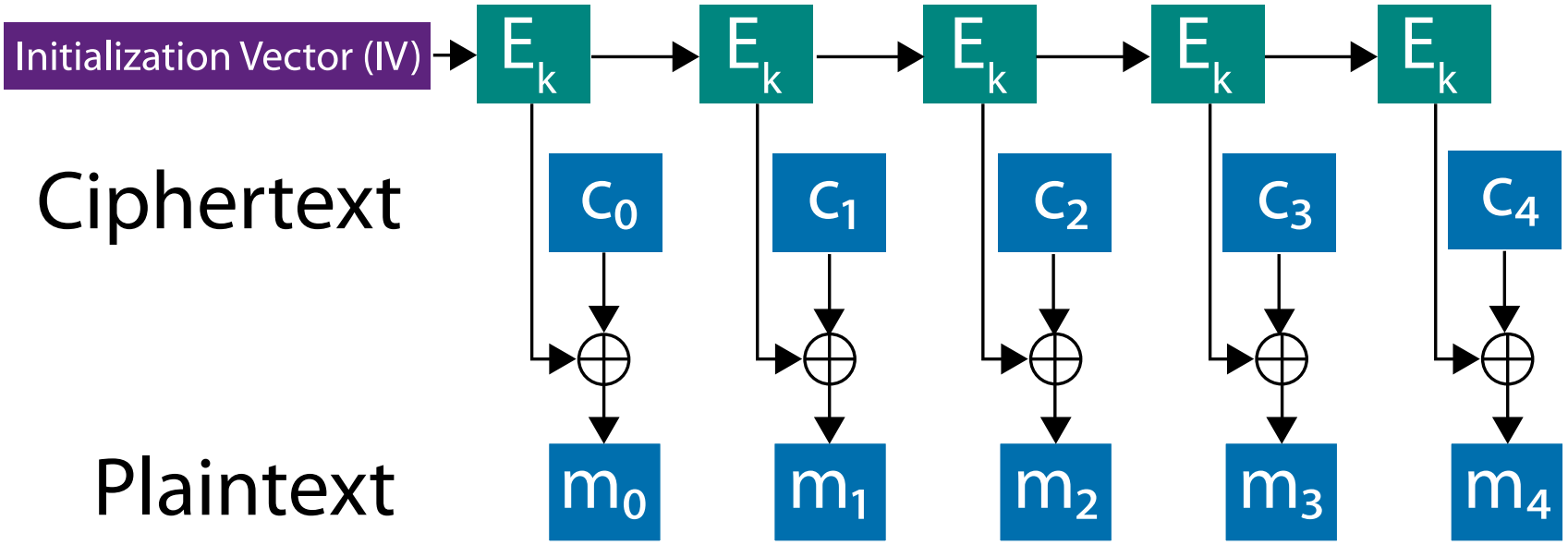
- Ciphertext errors only affect two plaintext blocks, one in a predictable way.
- Encryption must be done sequentially.
- Decryption can be random-access and is fully parallelisable.




CBC Decryption

# Output Feedback Mode (OFB)

**Output Feedback Mode (OFB)** effectively turns a block cipher into a synchronous stream cipher.



# OFB Properties

Identical plaintext results in identical ciphertext when the same plaintext is enciphered using the same key and IV.

**Chaining Dependencies**: (*Same as a stream cipher*) The key stream is plaintext independent.

**Error propagation**: (*Same as a stream cipher*) Bit errors in ciphertext blocks cause errors in the same position in the plaintext.

**Error recovery**: (*Same as a stream cipher*) Recovers from bit errors, but not bit loss
(misalignment of key stream)

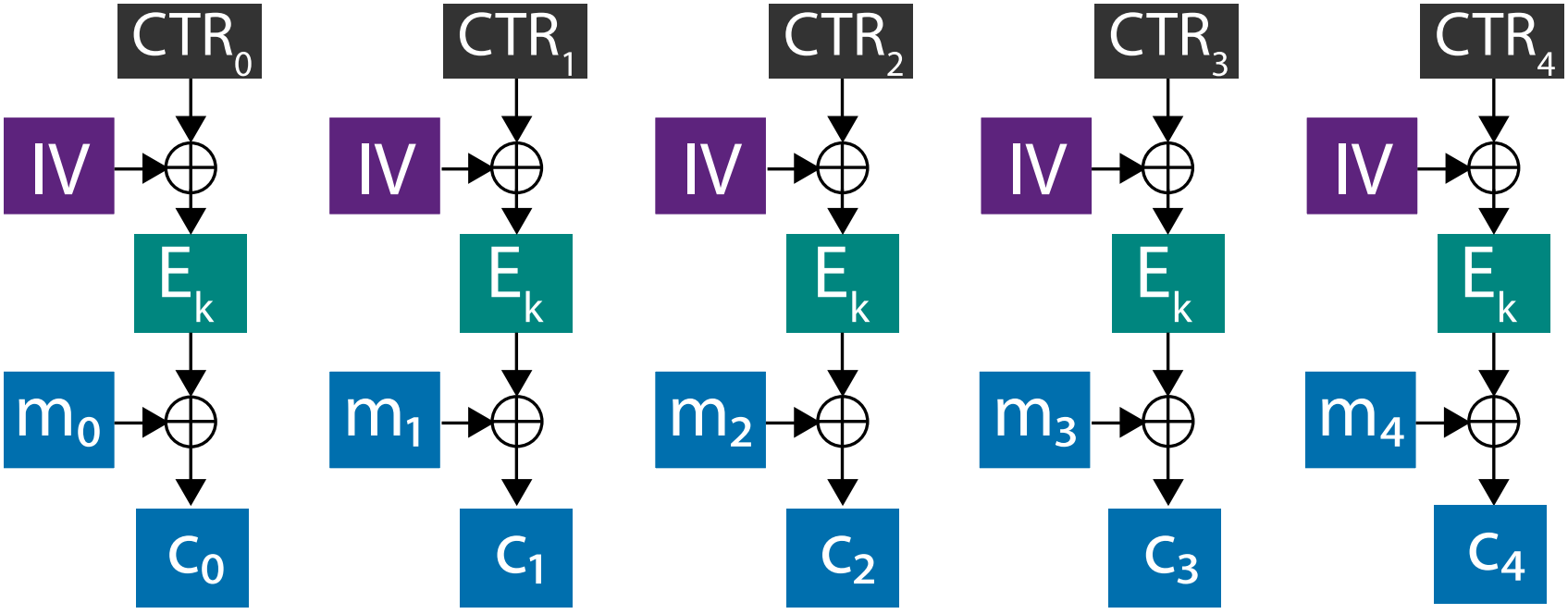
**Throughput**: Key stream may be calculated independently — e.g. pre-computed — before encryption/decryption become parallelisable.

**IV *must* change**: Otherwise it becomes a two time pad.

# Counter Mode (CTR)

**Counter Mode (CTR)** modifies the IV for each block using a predictable counter function, turning the block cipher into a stream cipher.

The counter can be any function (e.g. a PRNG), but it is commonly just an incrementing integer.




CTR Mode Encryption

# CTR Properties

Identical plaintext results in identical ciphertext when the same plaintext is enciphered using the same key and IV.

**Chaining Dependencies**: (*Same as a stream cipher*) The key stream is plaintext independent.

**Error propagation**: (*Same as a stream cipher*) Bit errors in ciphertext blocks cause errors in the same position in the plaintext.

**Error recovery**: (*Same as a stream cipher*) Recovers from bit errors, but not bit loss
(misalignment of key stream)

**Throughput**: Both encryption and decryption can be randomly accessed and/or parallelised: the best we could hope for.

**IV *must* change**: Otherwise it becomes a two time pad.

OFB and CTR share a lot of these properties, because they both make the block cipher act as a stream cipher.

# GCM Mode

**Galois/Counter Mode (GCM)** mode is not strictly a cipher mode of operation since it also provides *authentication*: assurance the ciphertext has not been tampered with.

- An extension of CTR mode.
- While encryption happens, the ciphertext blocks are combined into something like a MAC.
- Unlike HMAC, is parallelisable (you can't combine two HMACs into one larger one).
- Used for low-latency, high-throughput dedicated hardware applications (network packets).

GCM mode is an example of *authenticated encryption*.