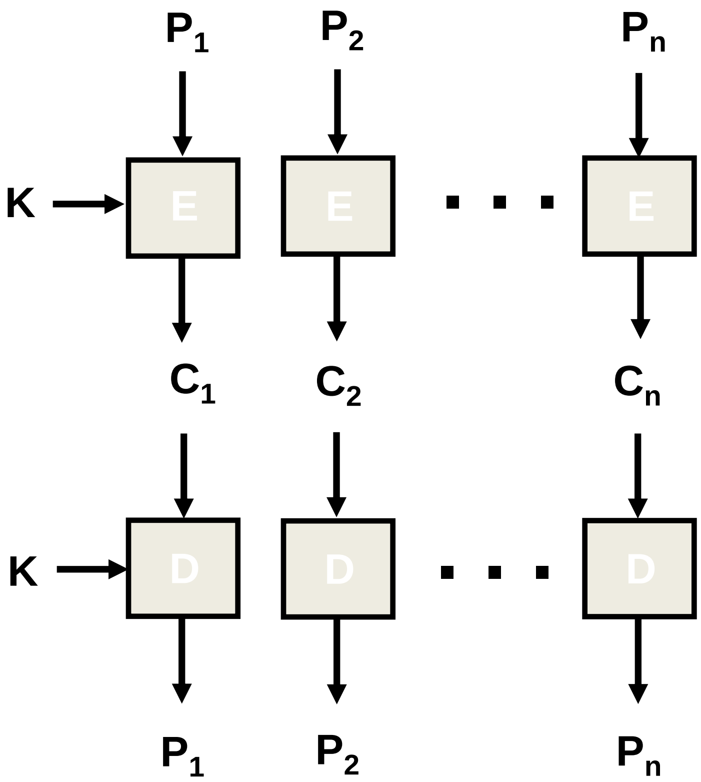
Week 6: Mode of Operation

[Recommendation for Block Cipher Modes of Operation NIST 800-38A, 2001]

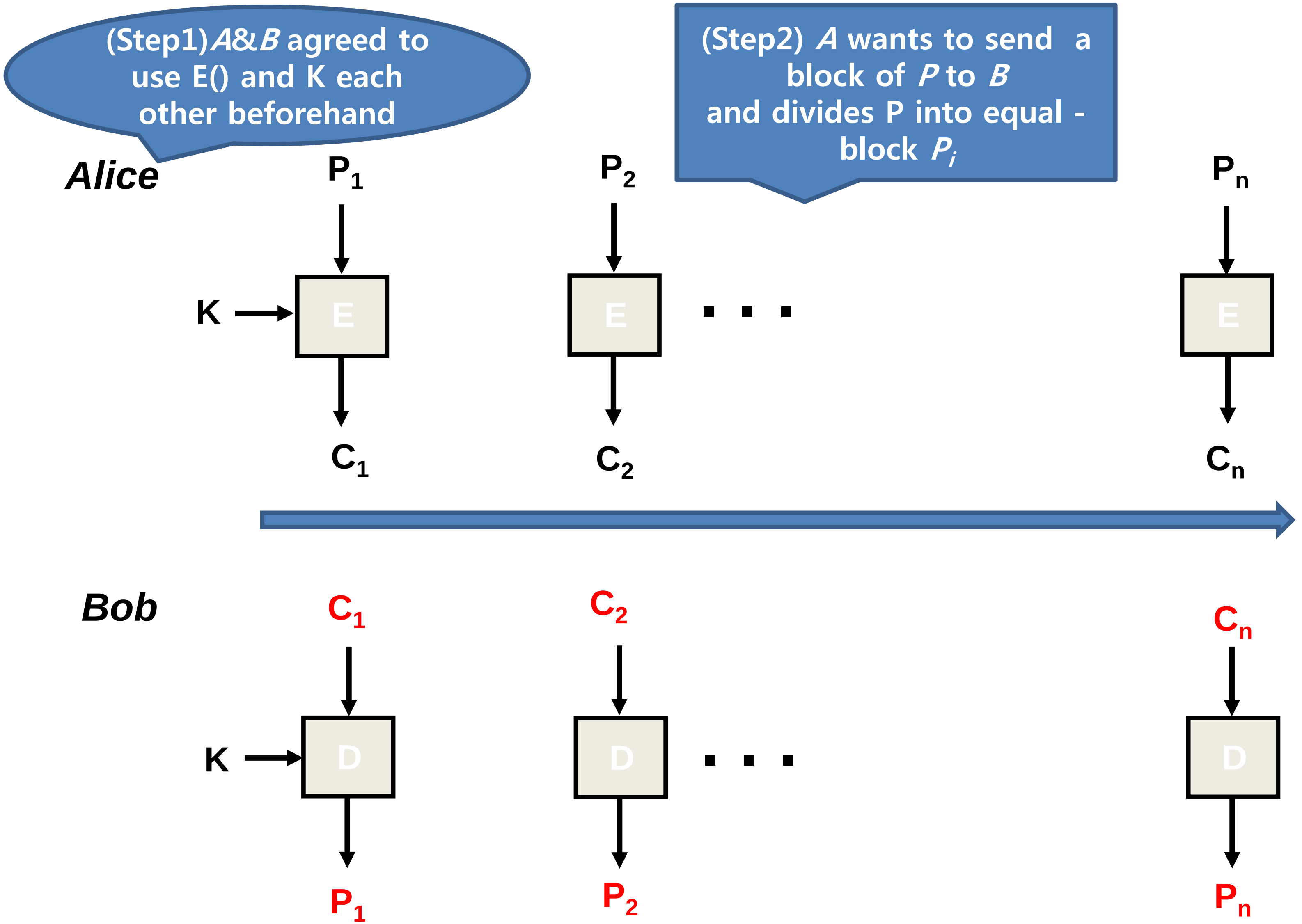
Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, NIST-800-38D,2007

Modes of Operation – ECB Mode(1/2)

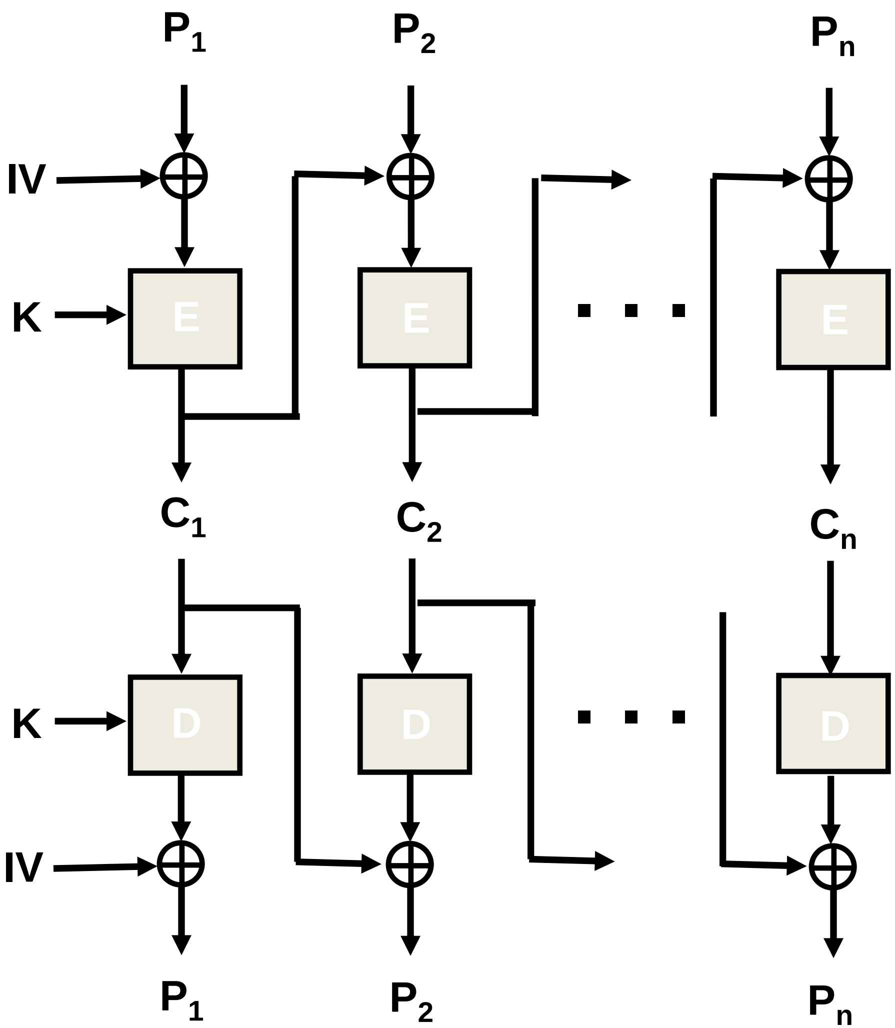


- **Electronic Code Book Mode**
  - ✓ Break a message into a sequence of plaintext blocks
  - ✓ Each plaintext block is encrypted (or decrypted) independently
  - ✓ The same plaintext block always produces the same ciphertext block
  - ✓ May not be secure; e.g., a highly structured message
  - ✓ Typically used for secure transmission of single vales (e.g., encryption key)

Modes of Operation – ECB Mode(2/2)



Modes of Operation – CBC Mode (1/2)



- *Cipher Block Chaining Mode*
  - ✓ Each ciphertext block is affected by previous blocks
  - ✓ No fixed relationship between the plaintext block and its input to the encryption function
  - ✓ The same plaintext block, if repeated, produces different ciphertext blocks
  - ✓ IV(Initializing Vector) must be known to both ends
  - ✓ Most widely used for block encryption

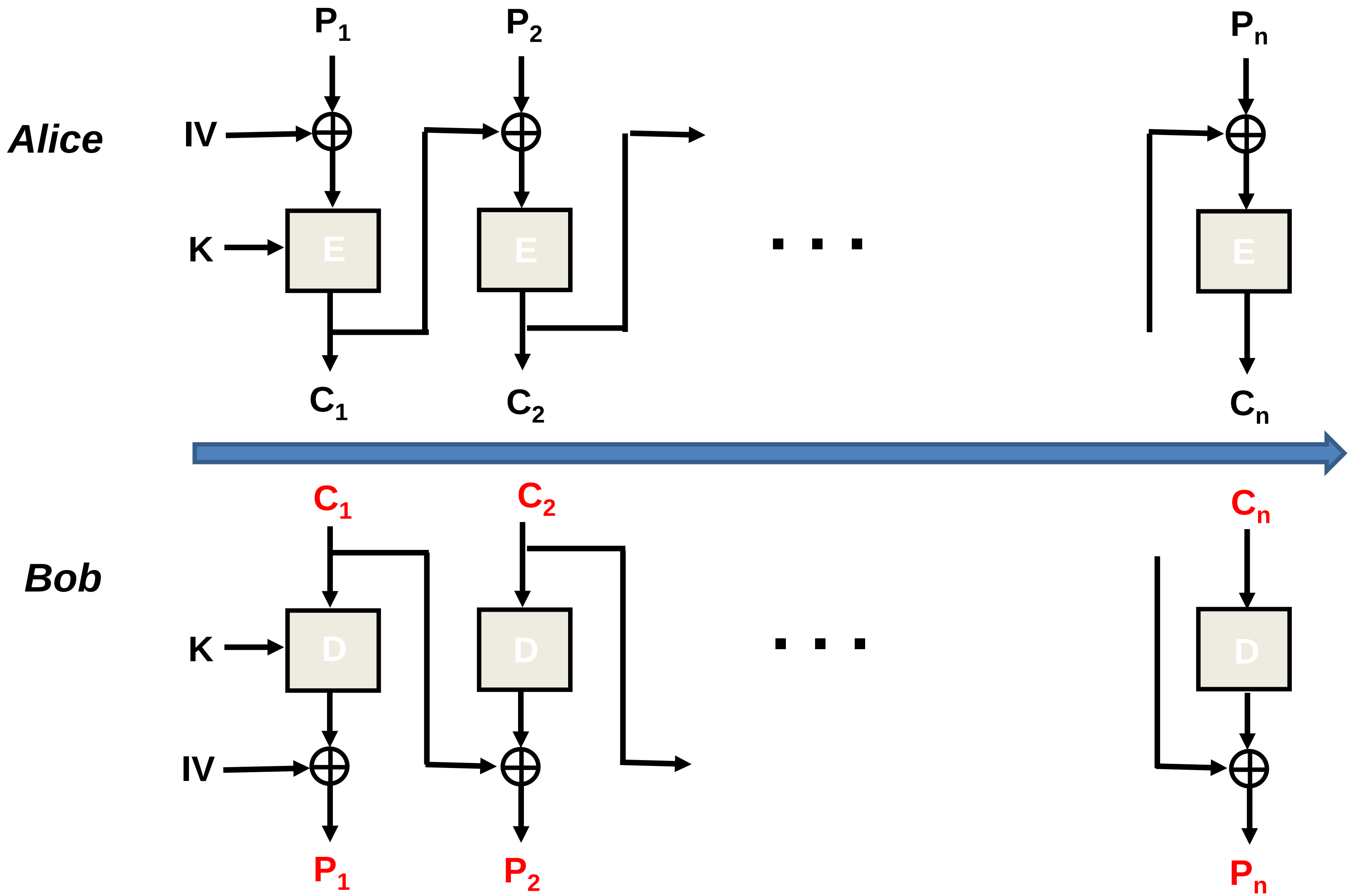
$C_1 = E_K(P_1 \oplus IV)$ $\quad$ $C_3 = E_K(P_3 \oplus C_2)$

$P_1 = IV \oplus D_K(C_1)$ $\quad$ $P_3 = C_2 \oplus D_K(C_3)$

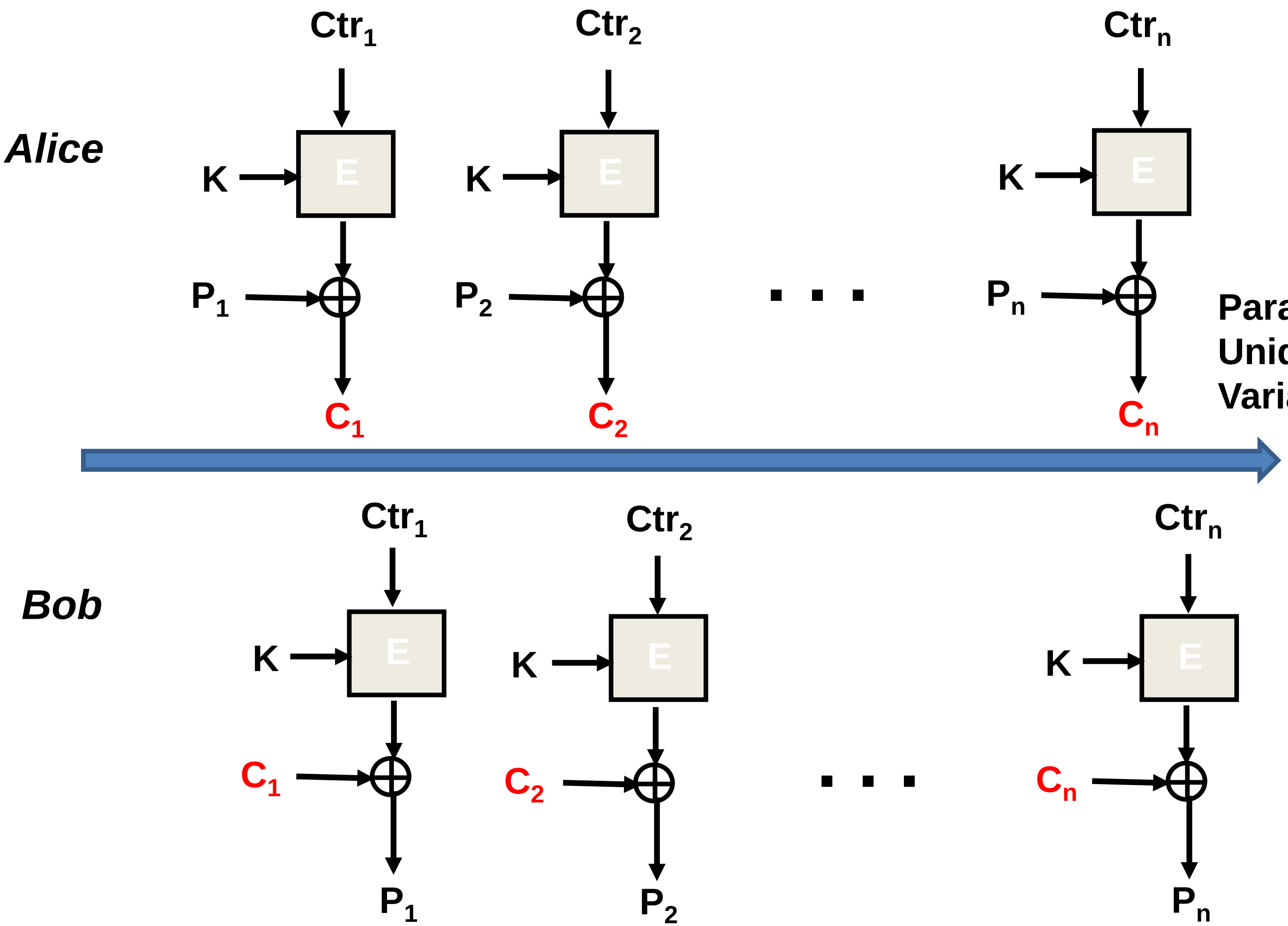
$C_2 = E_K(P_2 \oplus C_1)$ $\quad$ $C_4 = E_K(P_4 \oplus C_3)$

$P_2 = C_1 \oplus D_K(C_2)$ $\quad$ $P_4 = C_3 \oplus D_K(C_4)$

Modes of Operation – CBC Mode (2/2)



Modes of Operation – Ctr Mode



Parallel Operation
Unique Counter
Variation of CBC

Stream Cipher

❑ Overview

- Originate from one-time pad
- bit-by-bit Exor with pt and key stream ($c_i = m_i \oplus z_i$)
- Encryption = Decryption --> Symmetric
- Use LFSR (Linear Feedback Shift Register)
- (external) Synchronous or self-synchronous

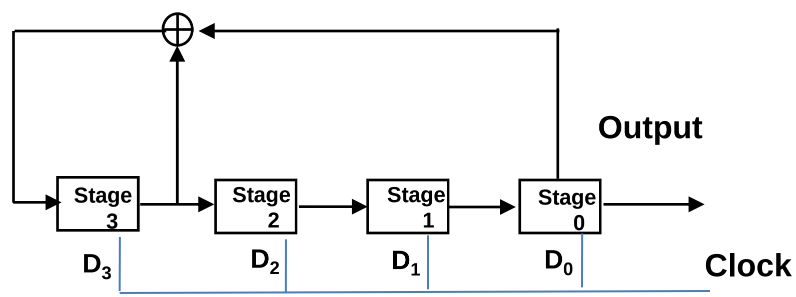
❑ Properties

- Faster and Low Complexity in H/W -> Lightweight !
- Security measure : Period of key stream,
  LC(Linear Complexity), Statistical properties
- Vast amounts of theoretical knowledge
- Proprietary and Confidential for Military

How LFSR works

- Notation: < L, C[D]> where connection polynomial
  $C[D] = 1 + c_1D + c_2D^2 + \ldots + c_LD^L \in Z_2[D]$
- If $c_L=1$, {i.e., deg{C[D]}=L}, C[D] is called a nonsingular polynomial
- If initial vector σ_0 is $[s_{L-1}, \ldots, s_1, s_0]$, $s_i = \{0,1\}$, output sequence s= $s_0, s_1, \ldots$ is uniquely determined by the recursion
  $s_j = (c_1 s_{j-1} + c_2 s_{j-2} + \ldots + c_L s_{j-L}) \bmod 2 , j \geq L$
- (Ex) $<4, 1 + D + D^4>$, $\sigma_0 = [0,1,1,0]$ ➔ $c_1 = 1$, $c_4 = 1$, $s_4 = s_3 + s_0$

t	D_3	D_2	D_1	D_0	t	D_3	D_2	D_1	D_0
0	0	1	1	0 (6)	8	1	1	1	0 (14)
1	0	0	1	1 (3)	9	1	1	1	1 (15)
2	1	0	0	1 (9)	10	0	1	1	1 (7)
3	0	1	0	0 (4)	11	1	0	1	1 (11)
4	0	0	1	0 (2)	12	0	1	0	1 (5)
5	0	0	0	1 (1)	13	1	0	1	0 (10)
6	1	0	0	0 (8)	14	1	1	0	1 (13)
7	1	1	0	0 (12)	15	0	1	1	0 (6)

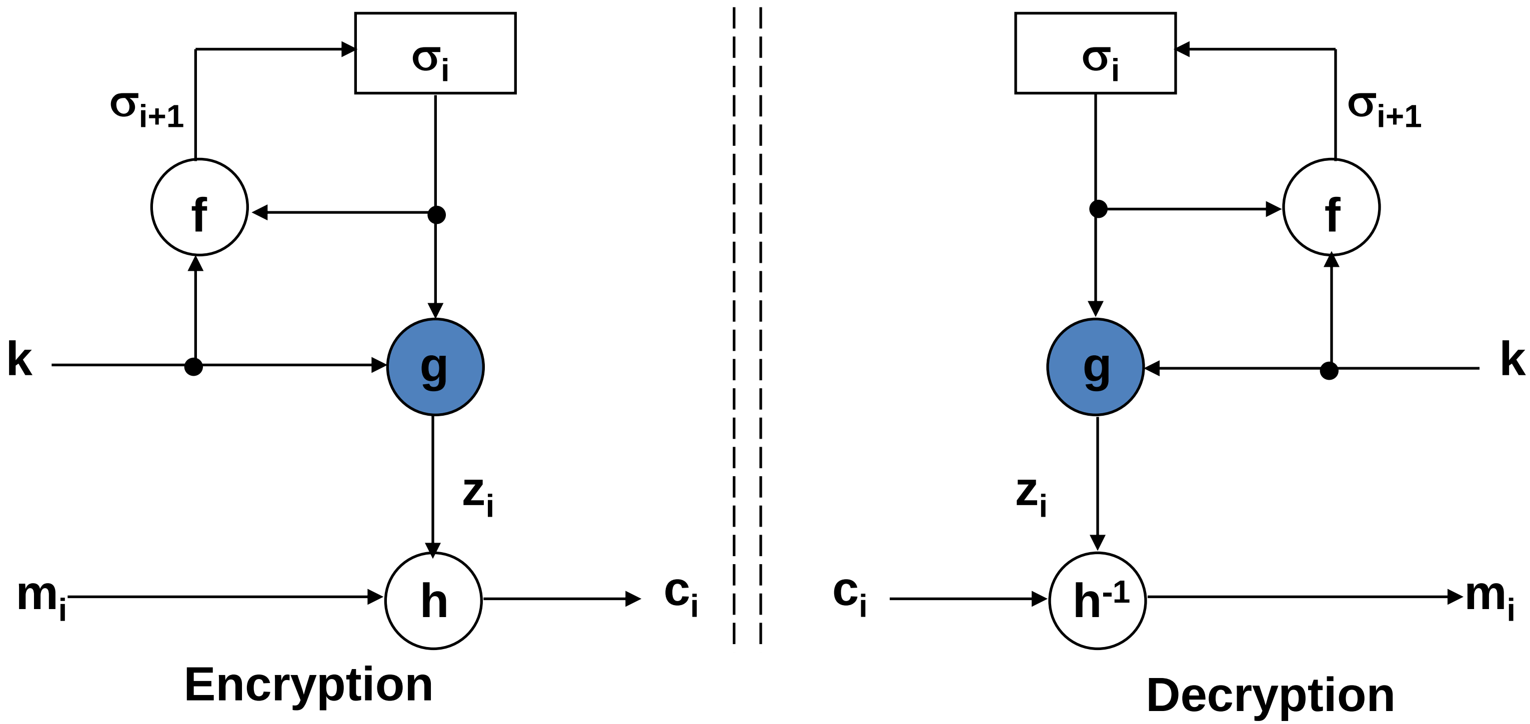


Output seq. = 0,1,1,0,0,1,0,0,0,1,1,1,1,0,1

15

Synchronous Stream Cipher(1/2)

- f : next state ft, $\sigma_{i+1} = f(\sigma_i , k)$, σ_0 : initial value
- g : keystream generating ft (e.g., LFSR or Block Cipher)
- $z_i = g(\sigma_i , k)$, k : key
- h : output ft, $c_i = h(z_i, m_i)$, m_i : pt, z_i : key stream, c_i:ct

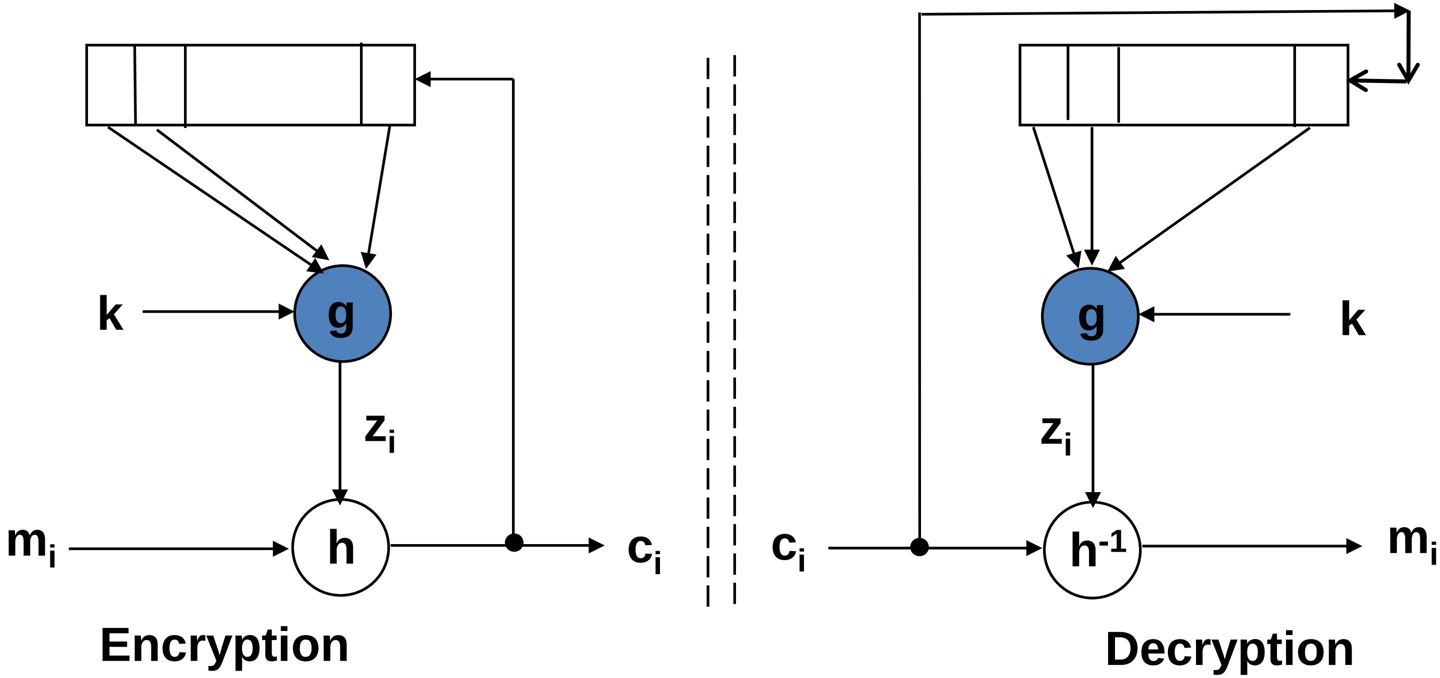


Synchronous Stream Cipher(2/2)

- Keystream is independent of pt and ct
- Properties
  - Synchronization requirement
  - No error propagation
  - Active attack
    - Insertion, deletion or replay will lose synchronization
    - Change selected ciphertext digits ➔ Need to have integrity check mechanisms

Self-Sync. Stream Cipher(1/2)

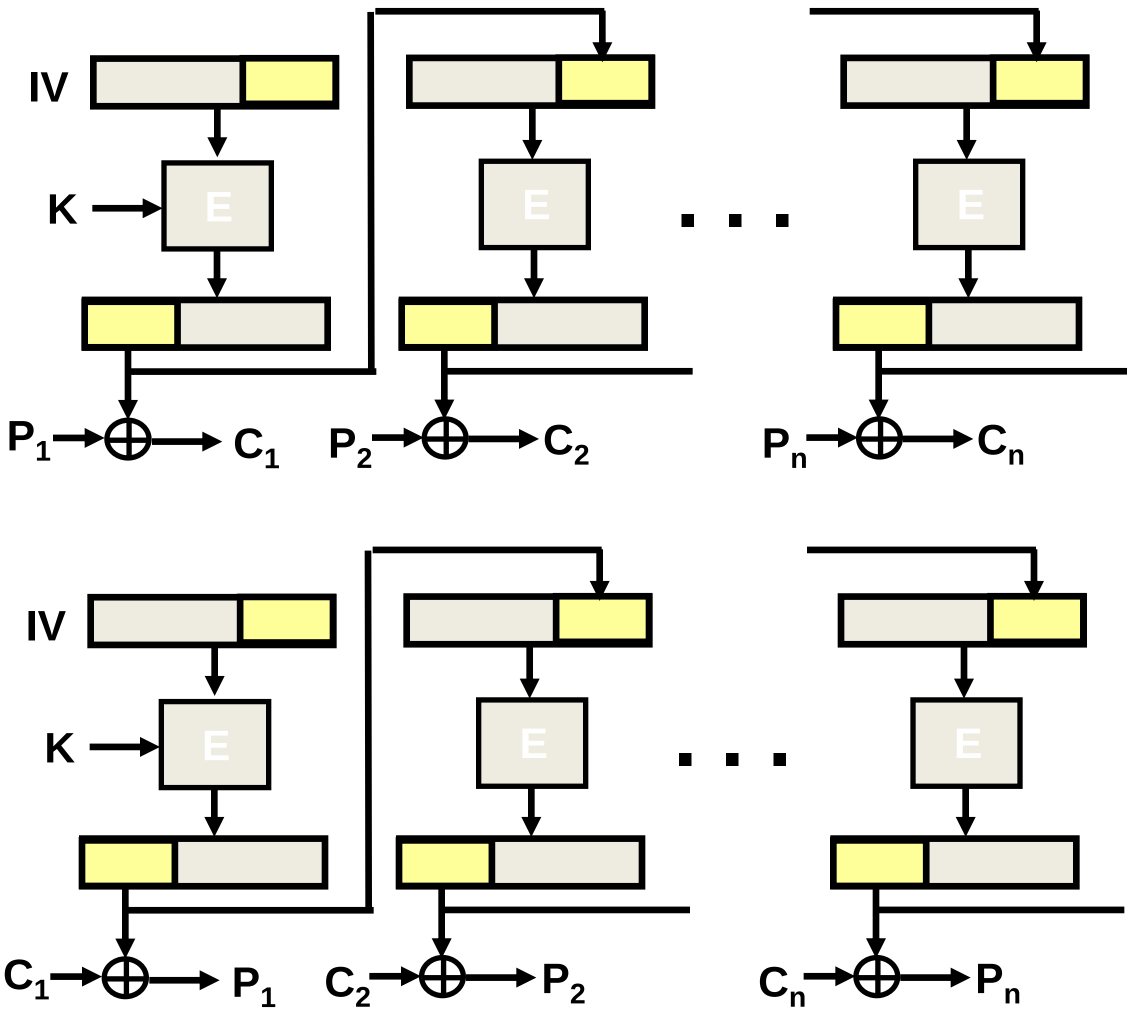
- $\sigma_i = (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1})$, $\sigma_0 = (c_{-t}, c_{-t+1}, \ldots, c_{-1})$: initial value
- g : keystream generating ft, (e.g., LFSR or Block Cipher)
- $z_i = g(\sigma_i, k)$, k : key
- h : output ft, $c_i = h(z_i, m_i)$, m_i : pt, z_i : keystream, c_i : ct



Self-Sync. Stream Cipher(2/2)

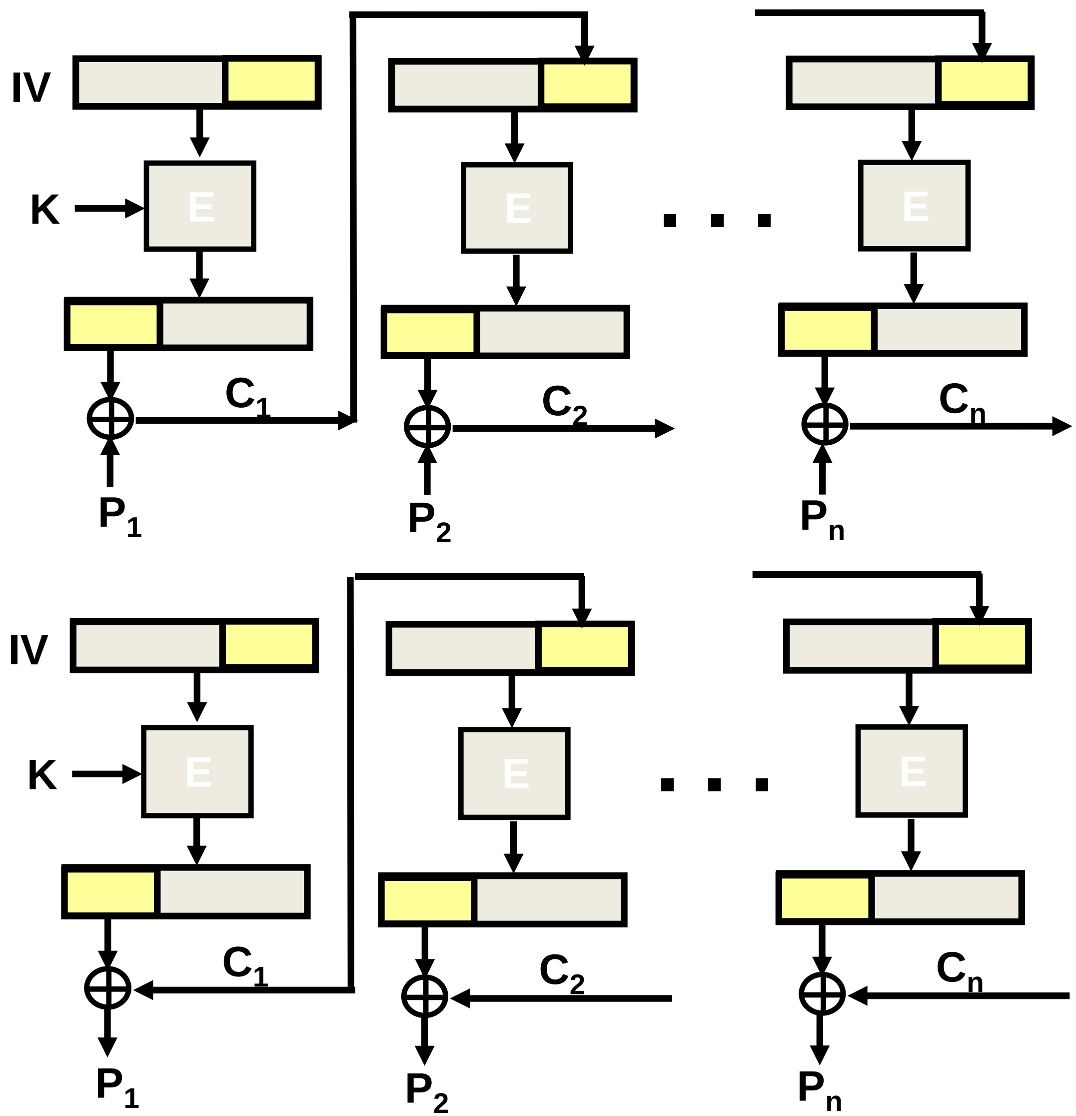
- Keystream is dependent of pt and ct
- Properties
  - Self-Synchronization
  - Limited error propagation
  - Active attack
    - Difficult to detect insertion, deletion, or replay
    - Easy to find passive modification
  - More diffusion ➔more resistant against attacks based on plaintext redundancy

OFB Mode = Sync. Stream cipher



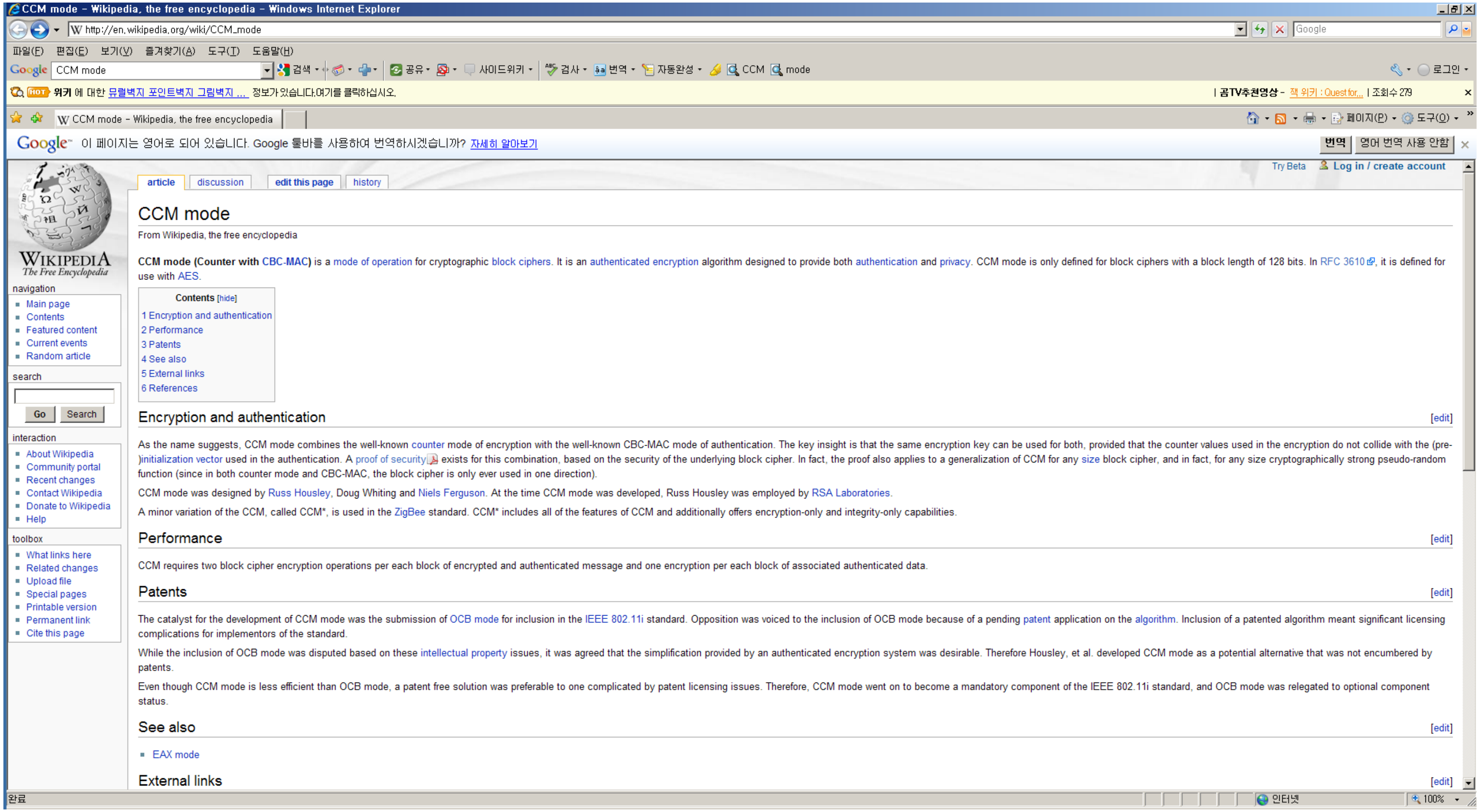
- ***Output Feedback Mode***
  - ✓ The structure is similar to that of CFB, but
    - CFB: Ciphertext is fed back to the shift register
    - OFB: Output of E is fed back to the shift register
  - ✓ For security reason, only the full feedback (j = block size) mode is used
  - ✓ No error propagation
  - ✓ More vulnerable to a message stream modification attack
  - ✓ May useful for secure transmission over noisy channel (e.g., satellite communication)

CFB Mode = Self-sync stream cipher



- *Cipher Feedback Mode*
  - ✓ A way of using a block cipher as a stream cipher
  - ✓ A shift register of block size maintains the current state of the cipher operation, initially set to some IV
  - ✓ The value of the shift register is encrypted using key K and the leftmost j bits of the output is XORed with j-bit plaintext P_i to produce j-bit ciphertext C_i
  - ✓ The value of the shift register is shifted left by j bits and the C_i is fed back to the rightmost j bits of the shift register
  - ✓ Typically j = 8, 16, 32, 64 …
  - ✓ Decryption function D_K is never used

Mode of Operation -CCM



CCM mode - Wikipedia, the free encyclopedia - Windows Internet Explorer

http://en.wikipedia.org/wiki/CCM_mode

article | discussion | edit this page | history

Try Beta | Log in / create account

navigation
- Main page
- Contents
- Featured content
- Current events
- Random article

search

Go | Search

interaction
- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Cite this page

CCM mode

From Wikipedia, the free encyclopedia

CCM mode (Counter with CBC-MAC) is a mode of operation for cryptographic block ciphers. It is an authenticated encryption algorithm designed to provide both authentication and privacy. CCM mode is only defined for block ciphers with a block length of 128 bits. In RFC 3610, it is defined for use with AES.

Contents [hide]

Encryption and authentication [edit]

As the name suggests, CCM mode combines the well-known counter mode of encryption with the well-known CBC-MAC mode of authentication. The key insight is that the same encryption key can be used for both, provided that the counter values used in the encryption do not collide with the (pre-)initialization vector used in the authentication. A proof of security exists for this combination, based on the security of the underlying block cipher. In fact, the proof also applies to a generalization of CCM for any size block cipher, and in fact, for any size cryptographically strong pseudo-random function (since in both counter mode and CBC-MAC, the block cipher is only ever used in one direction).

CCM mode was designed by Russ Housley, Doug Whiting and Niels Ferguson. At the time CCM mode was developed, Russ Housley was employed by RSA Laboratories.

A minor variation of the CCM, called CCM*, is used in the ZigBee standard. CCM* includes all of the features of CCM and additionally offers encryption-only and integrity-only capabilities.

Performance [edit]

CCM requires two block cipher encryption operations per each block of encrypted and authenticated message and one encryption per each block of associated authenticated data.

Patents [edit]

The catalyst for the development of CCM mode was the submission of OCB mode for inclusion in the IEEE 802.11i standard. Opposition was voiced to the inclusion of OCB mode because of a pending patent application on the algorithm. Inclusion of a patented algorithm meant significant licensing complications for implementors of the standard.

While the inclusion of OCB mode was disputed based on these intellectual property issues, it was agreed that the simplification provided by an authenticated encryption system was desirable. Therefore Housley, et al. developed CCM mode as a potential alternative that was not encumbered by patents.

Even though CCM mode is less efficient than OCB mode, a patent free solution was preferable to one complicated by patent licensing issues. Therefore, CCM mode went on to become a mandatory component of the IEEE 802.11i standard, and OCB mode was relegated to optional component status.

See also [edit]

- EAX mode

External links [edit]

Mode of operation - summary

- Use of mode
  - ECB : key management, useless for file encryption
  - CBC : File encryption, useful for MAC
  - *m*-bit CFB : self-sync, impossible to use channel with low BER
  - *m*-bit OFB : external-sync. *m*= 1, 8 or n
  - Ctr : secret ctr, parallel computation
  - CCM : authenticated encryption = ctr + CBC
  - Performance Degradation/ Cost Tradeoff