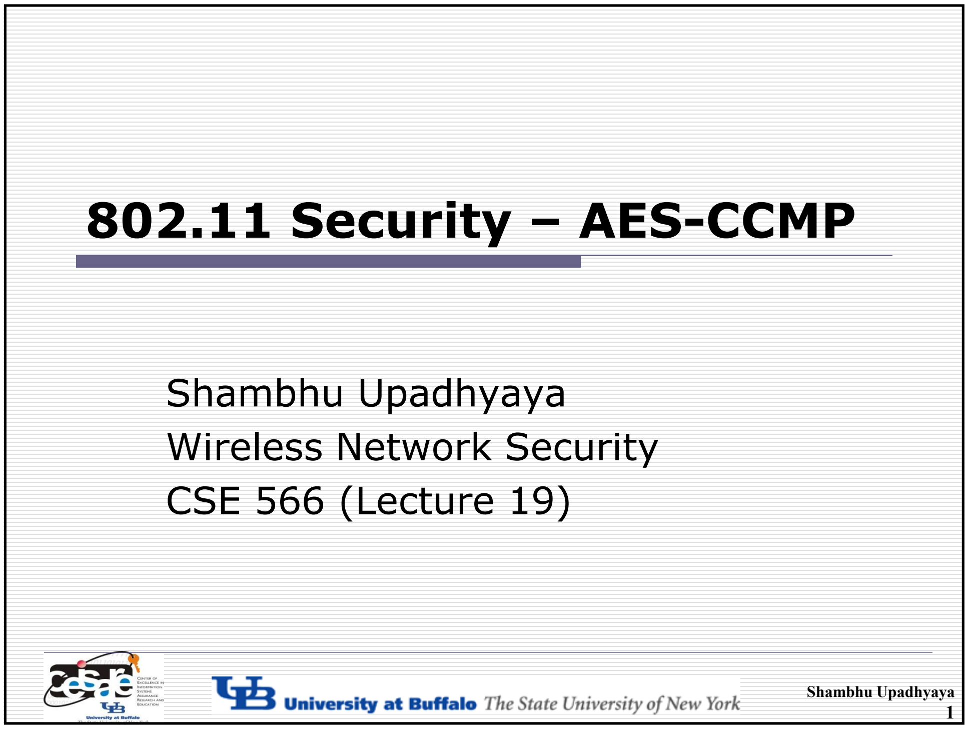
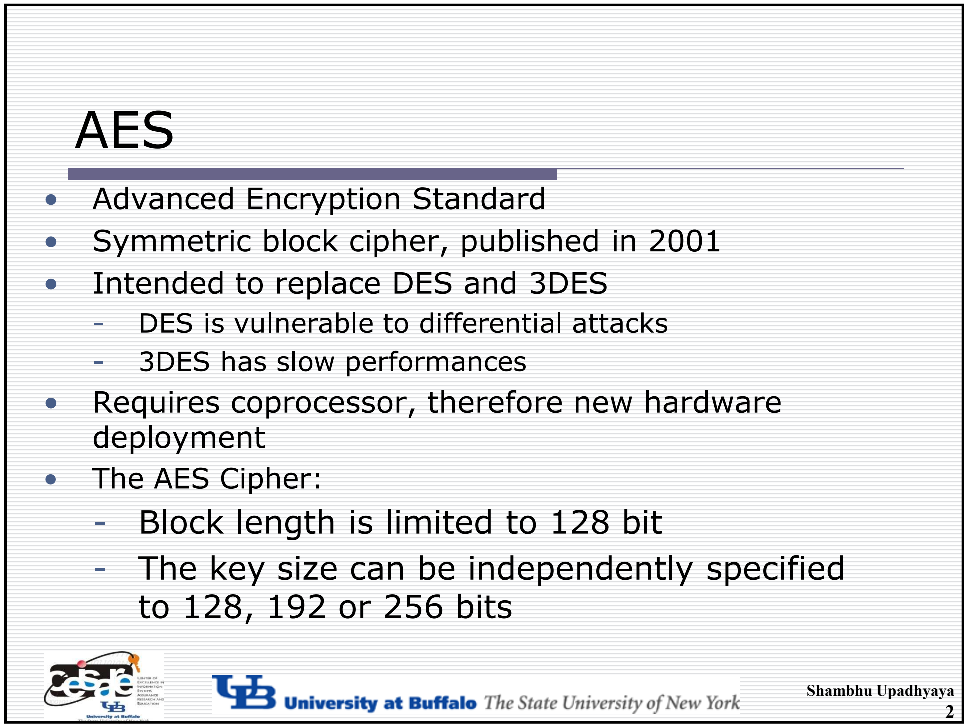


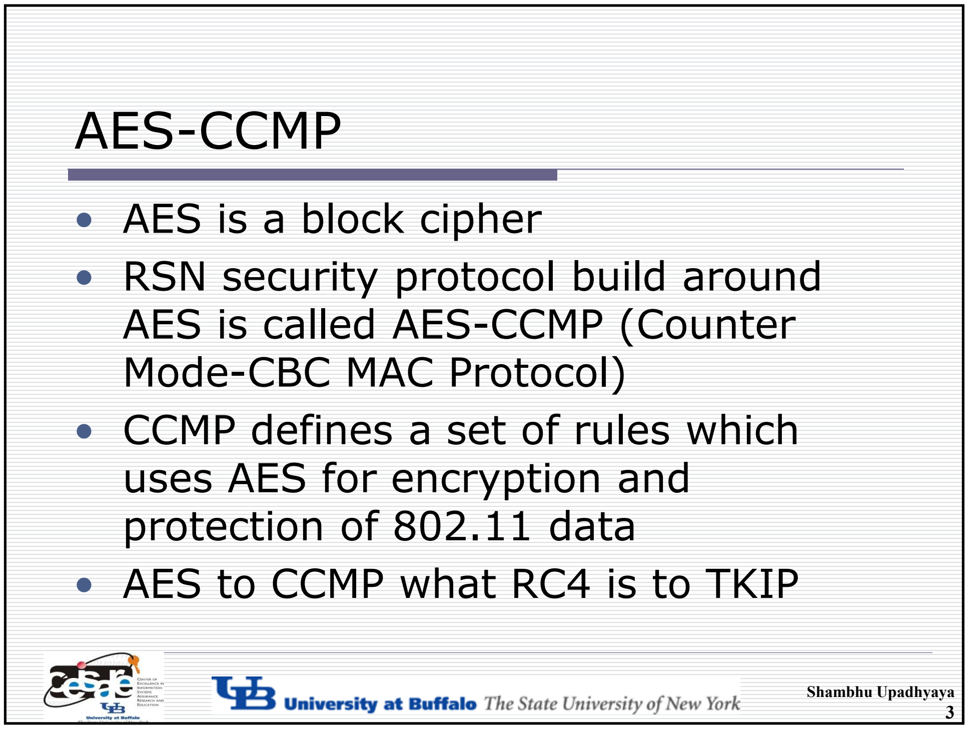
802.11 Security – AES-CCMP

Shambhu Upadhyaya
Wireless Network Security
CSE 566 (Lecture 19)



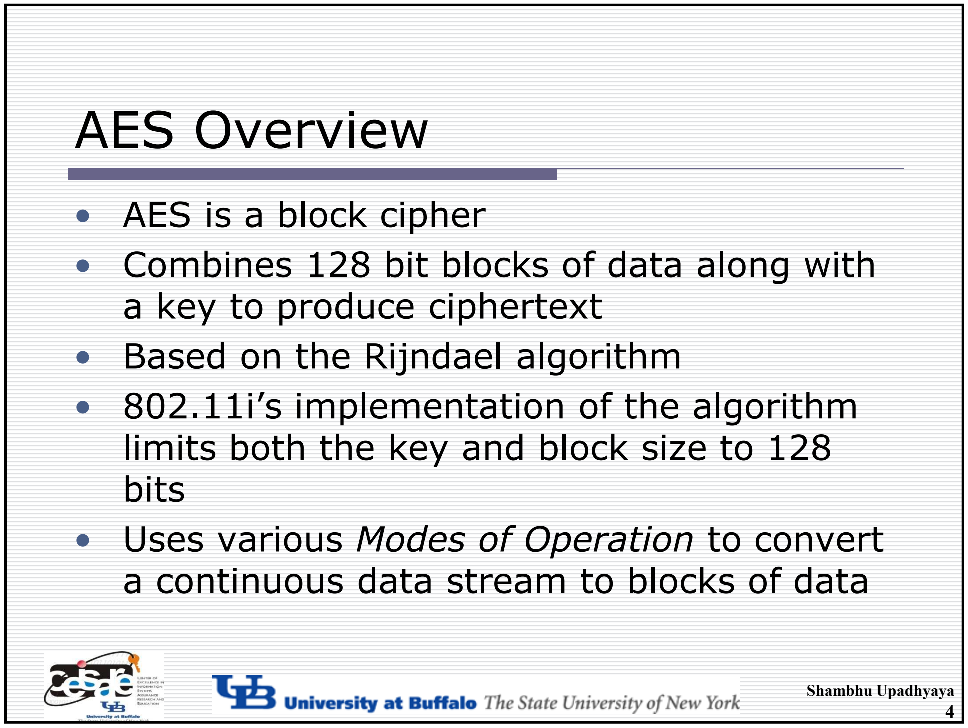
AES

- Advanced Encryption Standard
- Symmetric block cipher, published in 2001
- Intended to replace DES and 3DES
  - DES is vulnerable to differential attacks
  - 3DES has slow performances
- Requires coprocessor, therefore new hardware deployment
- The AES Cipher:
  - Block length is limited to 128 bit
  - The key size can be independently specified to 128, 192 or 256 bits



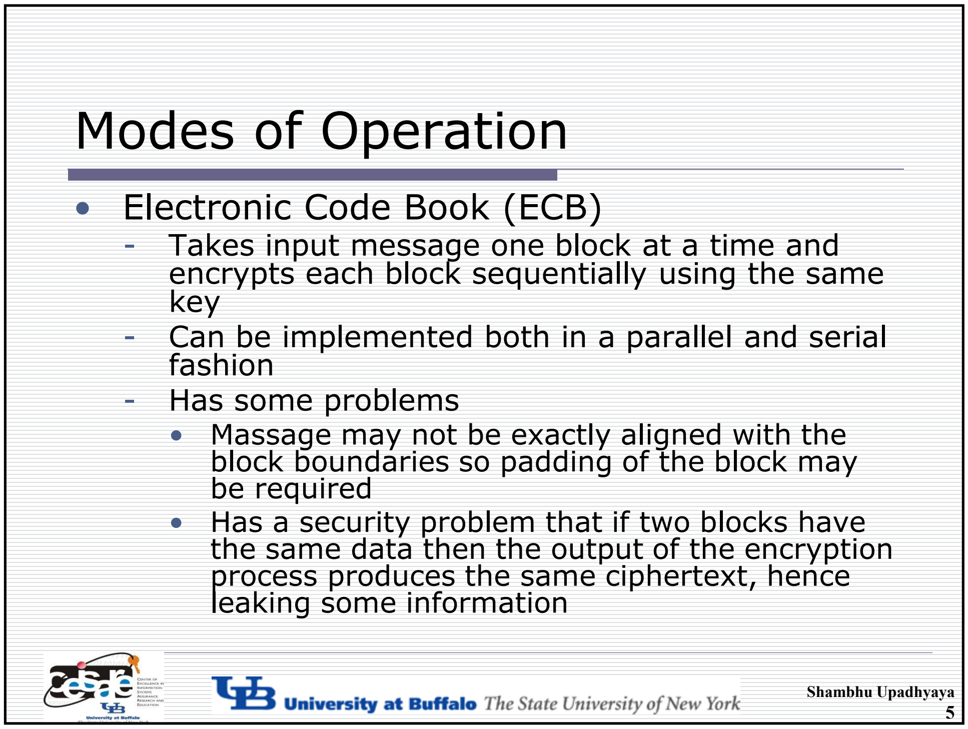
AES-CCMP

- AES is a block cipher
- RSN security protocol build around AES is called AES-CCMP (Counter Mode-CBC MAC Protocol)
- CCMP defines a set of rules which uses AES for encryption and protection of 802.11 data
- AES to CCMP what RC4 is to TKIP



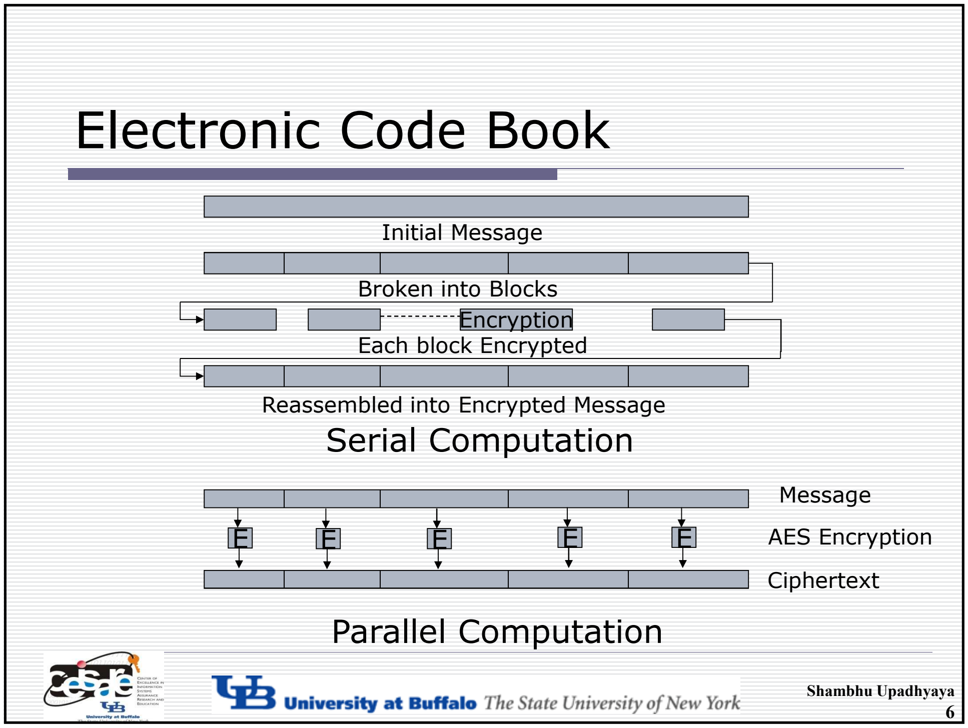
AES Overview

- AES is a block cipher
- Combines 128 bit blocks of data along with a key to produce ciphertext
- Based on the Rijndael algorithm
- 802.11i’s implementation of the algorithm limits both the key and block size to 128 bits
- Uses various *Modes of Operation* to convert a continuous data stream to blocks of data



Modes of Operation

- Electronic Code Book (ECB)
  - Takes input message one block at a time and encrypts each block sequentially using the same key
  - Can be implemented both in a parallel and serial fashion
  - Has some problems
    - Massage may not be exactly aligned with the block boundaries so padding of the block may be required
    - Has a security problem that if two blocks have the same data then the output of the encryption process produces the same ciphertext, hence leaking some information



Electronic Code Book

Initial Message

Broken into Blocks

Encryption

Each block Encrypted

Reassembled into Encrypted Message

Serial Computation

Message

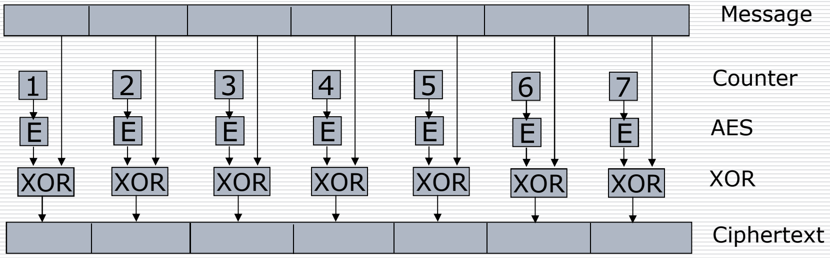
AES Encryption

Ciphertext

Parallel Computation

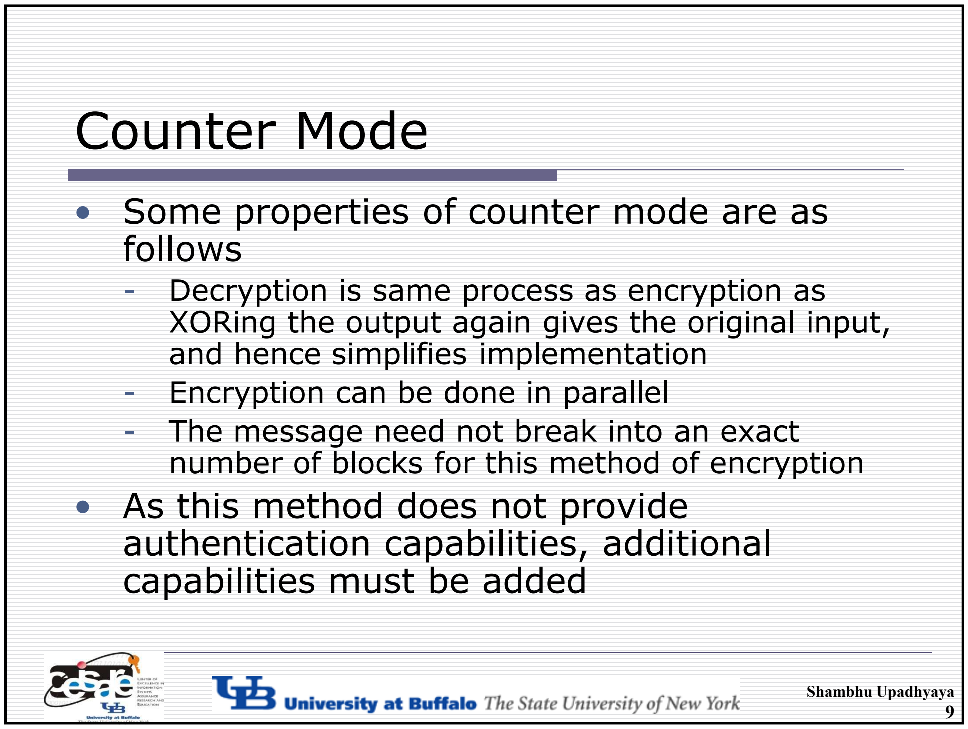
Counter Mode

- Does not use the AES cipher directly to encrypt the data
- Instead, it encrypts an arbitrary value called counter and XORs it with data to produce the ciphertext



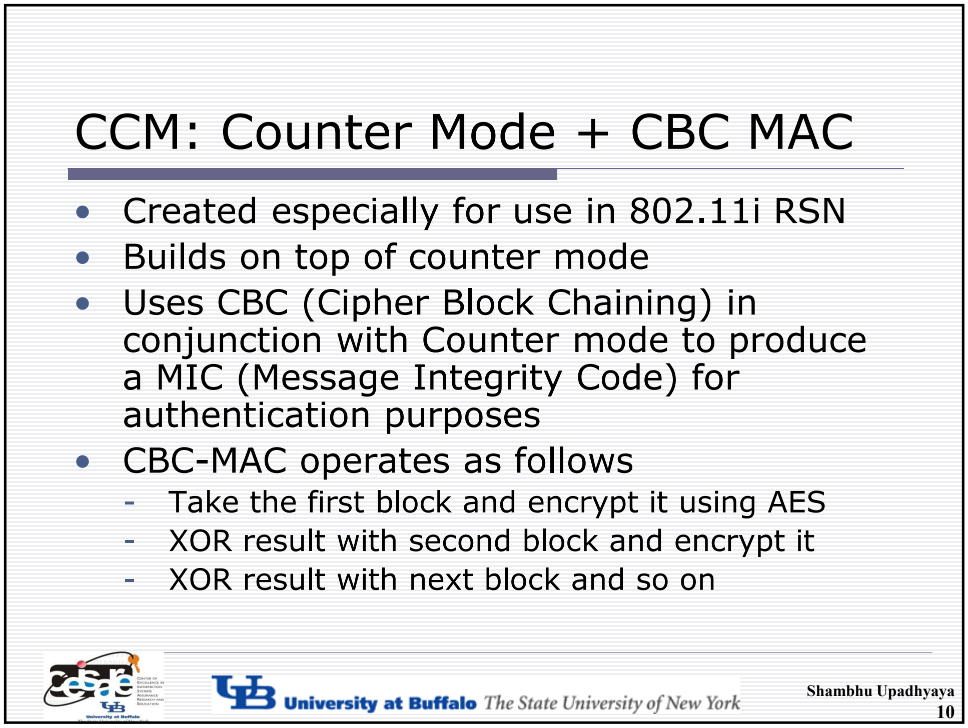
Counter Mode

- The counter might start at an arbitrary value and increment according to some pattern known to both the sender and receiver
- Because the counter changes for each block the problem of repeating blocks seen in ECB is avoided
- However, it would still encrypt two identical but separate messages identically
- To avoid this problem the counter is based on a nonce value rather than starting it from a fixed value



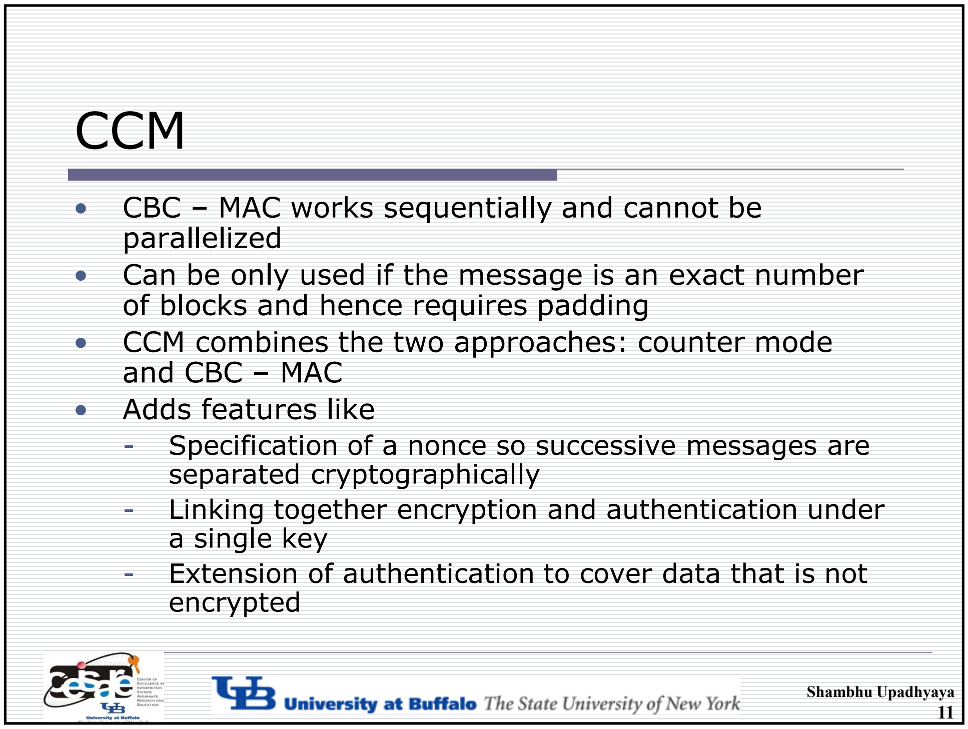
Counter Mode

- Some properties of counter mode are as follows
  - Decryption is same process as encryption as XORing the output again gives the original input, and hence simplifies implementation
  - Encryption can be done in parallel
  - The message need not break into an exact number of blocks for this method of encryption
- As this method does not provide authentication capabilities, additional capabilities must be added



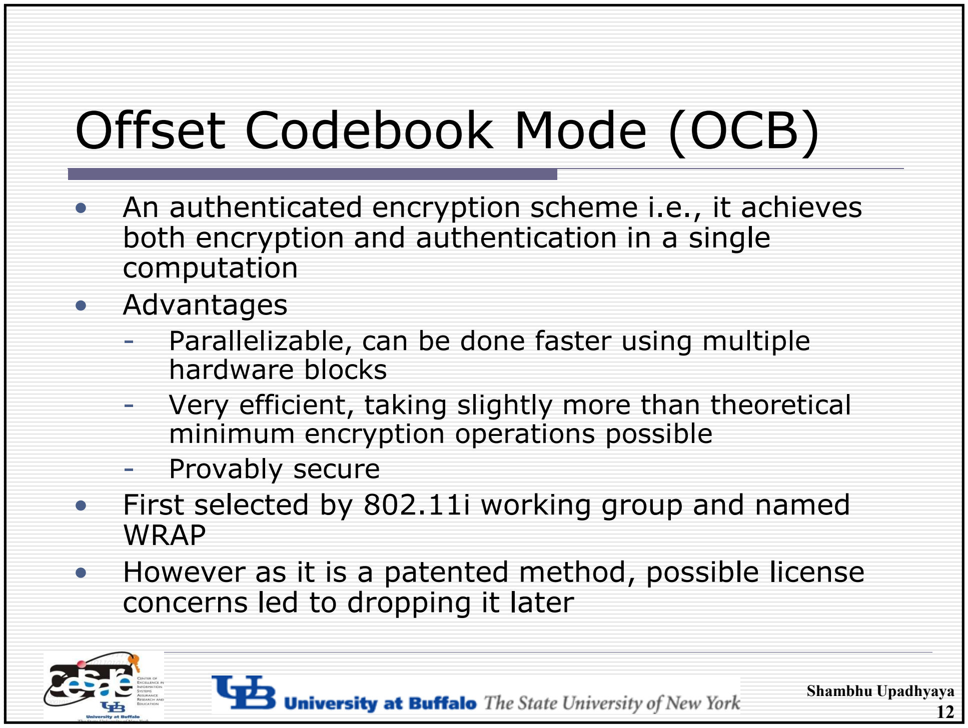
CCM: Counter Mode + CBC MAC

- Created especially for use in 802.11i RSN
- Builds on top of counter mode
- Uses CBC (Cipher Block Chaining) in conjunction with Counter mode to produce a MIC (Message Integrity Code) for authentication purposes
- CBC-MAC operates as follows
  - Take the first block and encrypt it using AES
  - XOR result with second block and encrypt it
  - XOR result with next block and so on



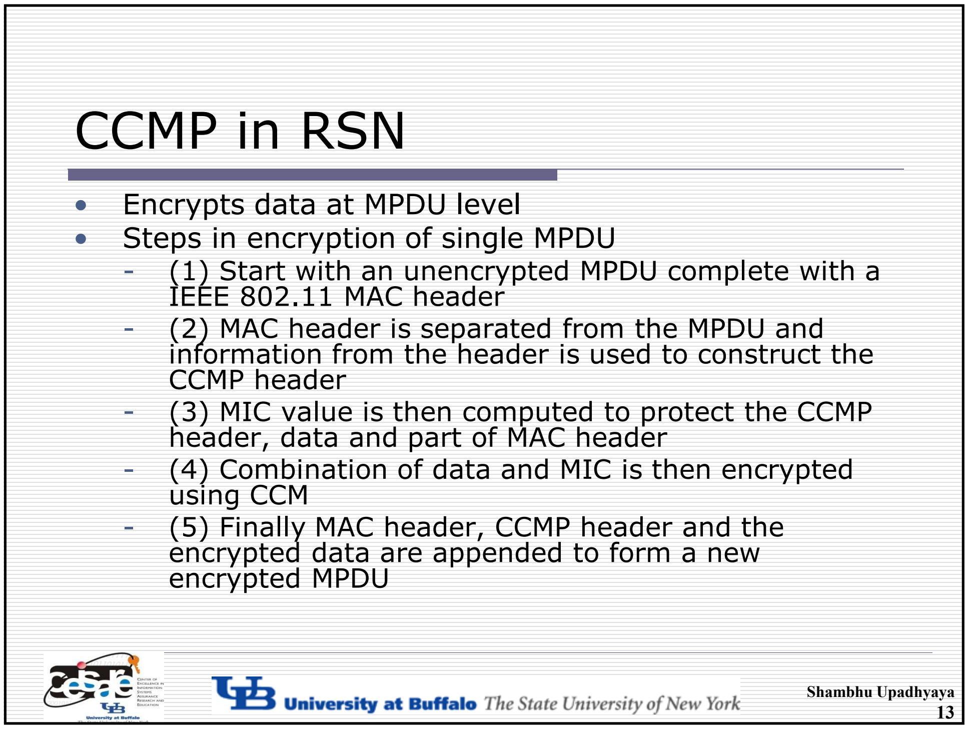
CCM

- CBC – MAC works sequentially and cannot be parallelized
- Can be only used if the message is an exact number of blocks and hence requires padding
- CCM combines the two approaches: counter mode and CBC – MAC
- Adds features like
  - Specification of a nonce so successive messages are separated cryptographically
  - Linking together encryption and authentication under a single key
  - Extension of authentication to cover data that is not encrypted



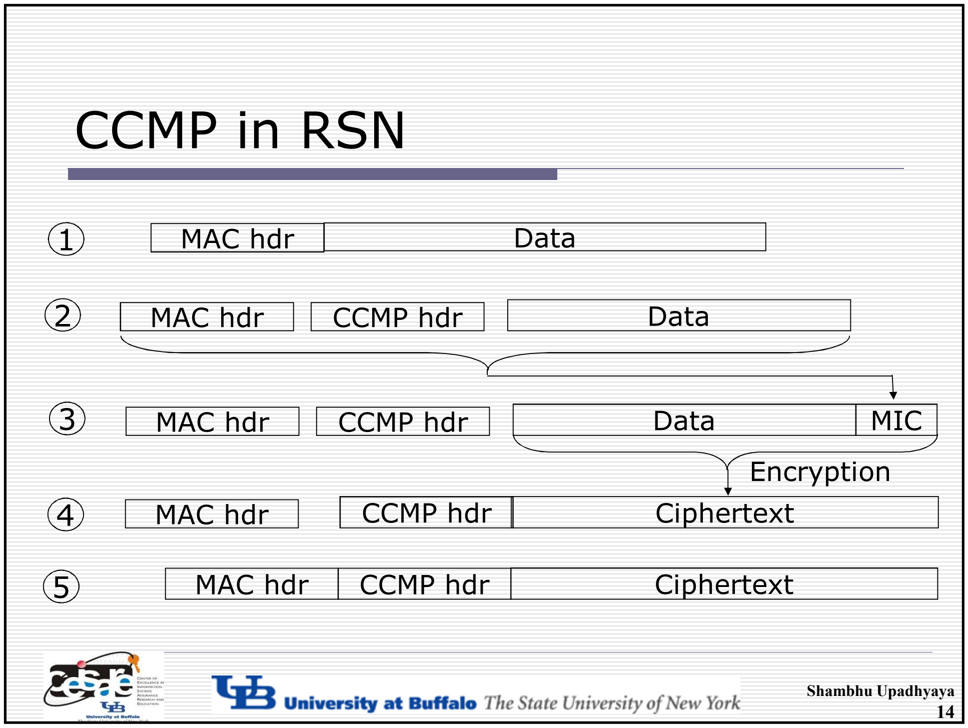
Offset Codebook Mode (OCB)

- An authenticated encryption scheme i.e., it achieves both encryption and authentication in a single computation
- Advantages
  - Parallelizable, can be done faster using multiple hardware blocks
  - Very efficient, taking slightly more than theoretical minimum encryption operations possible
  - Provably secure
- First selected by 802.11i working group and named WRAP
- However as it is a patented method, possible license concerns led to dropping it later



CCMP in RSN

- Encrypts data at MPDU level
- Steps in encryption of single MPDU
  - (1) Start with an unencrypted MPDU complete with a IEEE 802.11 MAC header
  - (2) MAC header is separated from the MPDU and information from the header is used to construct the CCMP header
  - (3) MIC value is then computed to protect the CCMP header, data and part of MAC header
  - (4) Combination of data and MIC is then encrypted using CCM
  - (5) Finally MAC header, CCMP header and the encrypted data are appended to form a new encrypted MPDU

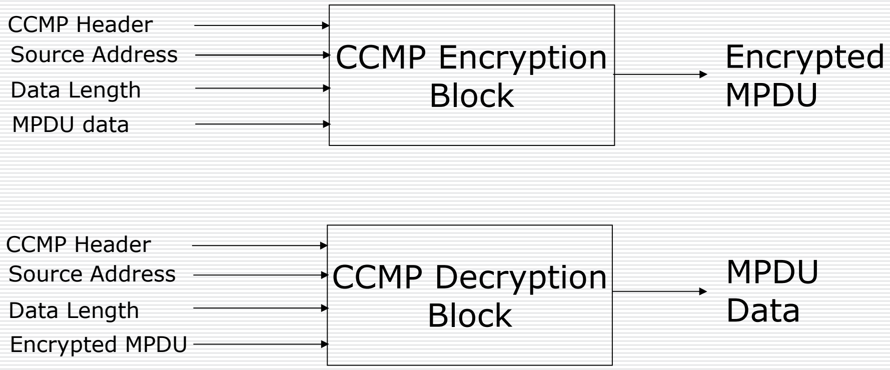


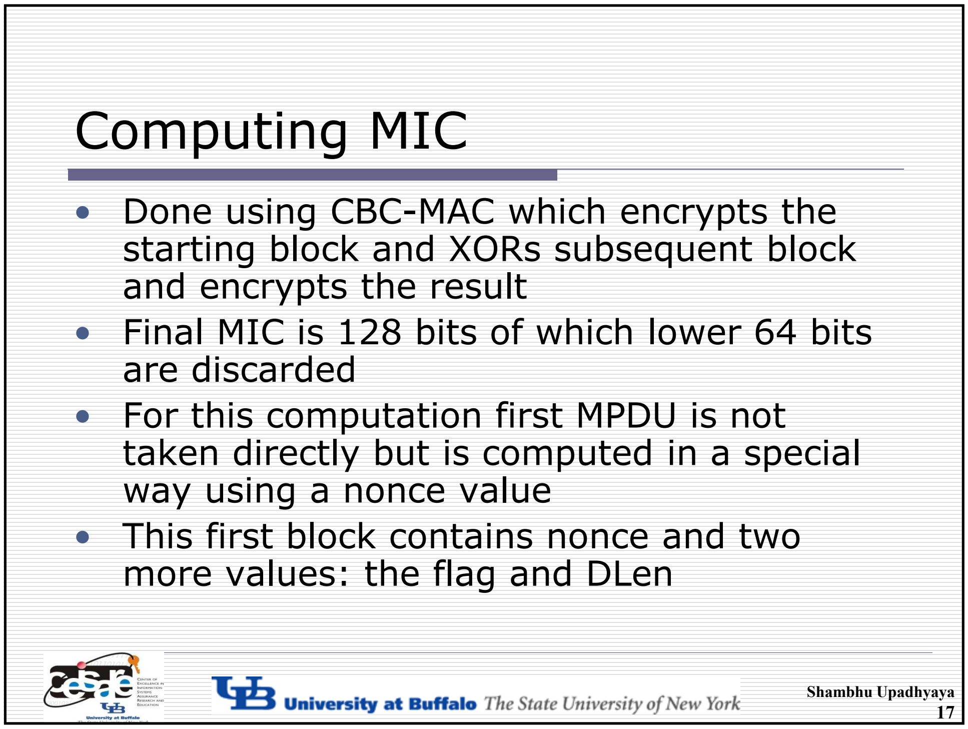
CCMP in RSN

CCMP Header

- Prepended to the encrypted data and transmitted in clear
- Has 2 purposes
  - Provides a 48 bit packet number (PN) for replay protection
  - In case of multicasts specifies the group key to be used
- Format of the header
  - 48 bits PN value
  - 1 byte is reserved
  - Rest is used for KeyID

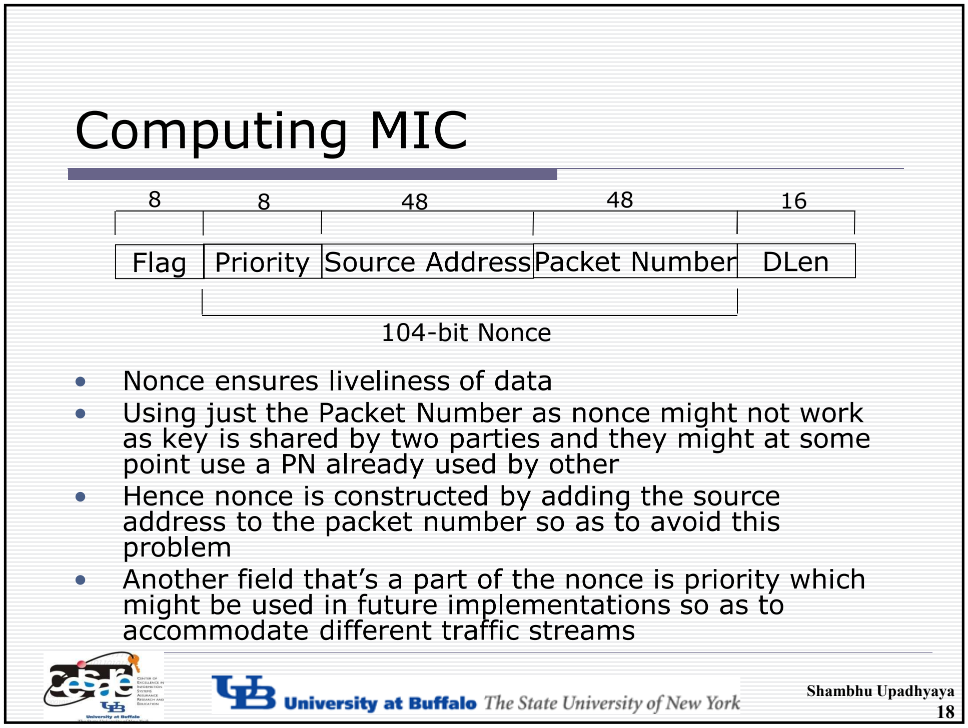
Implementation





Computing MIC

- Done using CBC-MAC which encrypts the starting block and XORs subsequent block and encrypts the result
- Final MIC is 128 bits of which lower 64 bits are discarded
- For this computation first MPDU is not taken directly but is computed in a special way using a nonce value
- This first block contains nonce and two more values: the flag and DLen



Computing MIC

8	8	48	48	16
Flag	Priority	Source Address	Packet Number	DLen

104-bit Nonce

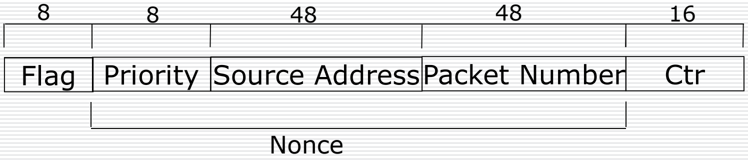
- Nonce ensures liveliness of data
- Using just the Packet Number as nonce might not work as key is shared by two parties and they might at some point use a PN already used by other
- Hence nonce is constructed by adding the source address to the packet number so as to avoid this problem
- Another field that’s a part of the nonce is priority which might be used in future implementations so as to accommodate different traffic streams

Computing MIC

- Flag field is 01011001 to specify, among other things, that MIC is 64 bits
- DLen indicates the size of the plaintext field
- As CBC-MAC works on blocks of fixed length, both the CCMP Header and Plaintext data need to be padded to get it to the required length
- MIC is computed across a combination of the special first block, the authenticated data, and plaintext including the zero padded bytes

Encrypting MPDUs

- After the MIC is computed the MPDU is encrypted using counter mode
- The counter is initialized in a way to avoid generating the same start value twice
- The Ctr value starts from 1 and counts upwards



8	8	48	48	16
Flag	Priority	Source Address	Packet Number	Ctr
	Nonce			

Decrypting MPDUs

- Firstly the correct pairwise keys are selected based on source MAC address
- The receiver checks the Packet Number of the received packet and ensures that it is not less than or equal to the last received PN
- If the Packet Number matches, the sequence number is combined with the source MAC and priority to create the nonce
- Then decryption proceeds as encryption, where successive values of counter are encrypted and XORed with received MPDUs
- Then the MIC is verified by recalculating the MIC over the data and padded zeroes
- If the new calculated value matched the MIC sent along with the MPDU the frame is accepted

References

- Jon Edney and William Arbaugh, Real 802.11 Security, Addison-Wesley, 2004