Block Cipher Confidentiality Modes (of Operation) Part 2: Block Modes Implemented as Stream Ciphers (CFB, OFB, CTR)

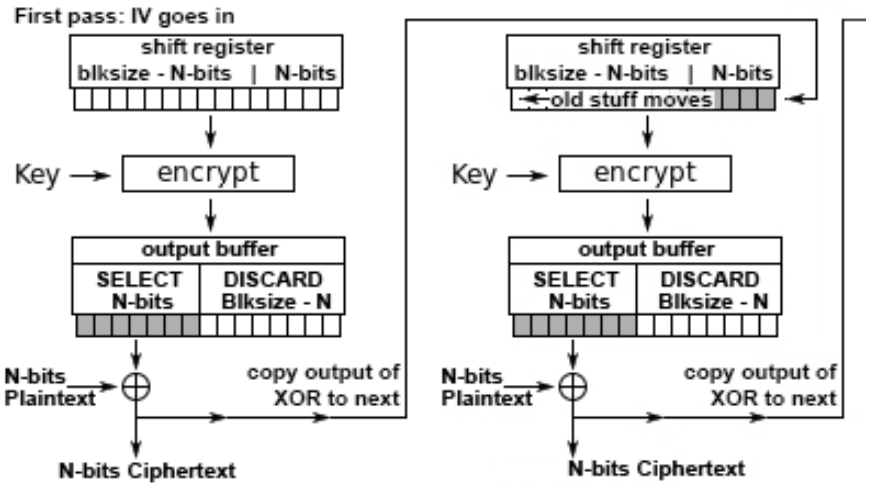
CFB - Cipher Feedback mode

A block mode implemented as a self-synchronizing stream cipher

- Allows encryption of partial blocks rather than requiring full blocks for encryption (padding not necessary)
- *Often you will see options for CFB, CFB-1, and CFB-8;* 8-bit blocks allow for one ascii character at a time
- Remember that the number of bits there is for the mode itself, separate from how many the block cipher employs.

The example below shows using 8-bit CFB, working with a 64-bit block algorithm

- The rule is, any size of *n*-bits where the *n* is less than or equal to the block size (in this case n=8bits)
- CFB uses a shift register queue the size of the block size (64 bits)



- First, an IV is put in the queue (like CBC)
- Queue is encrypted with the key, and the contents dumped into an output buffer.
- Leftmost 8 bits of output buffer are for use, the remainder is discarded.
- The selected 8 bits are XORed with 8 bits of the plaintext to output 8-bits of ciphertext.
- This ciphertext can now be transmitted, a copy is also pushed into the shift register, moving the pre-existing bits up in the queue.
- Bits left in the output buffer are dumped.
- The next plaintext bits are encrypted in the same manner, and it continues

- Plaintext patterns are concealed; input to the block cipher is randomized.
- Ciphertext is the same size as the plaintext, not counting the IV.
- More than one message can be encrypted with the same key provided that a different IV is used.
- An attacker with some of the plaintext can toggle bits in a given block and decrypt
- if IV is not unique for each message "session" the plaintext is vulnerable. Can be any incrementing number that doesn't repeat (like serial numbers). Data in storage could even use a function of the data lookup index.
- Self-recovering: A single bit error in ciphertext causes single error in plaintext, but also corrupts the shift register until it flushes out. Where m=block size, in n-bit CFG, a single ciphertext error will affect m/n-1 blocks (in 8-bit mode, 9 bytes garbled by a single bit ciphertext error). Also applies to synchronization. (see *self-synchronizing stream ciphers).*
- 1-bit CFB can recover from the addition or loss of single bits.
- Unlike other modes, synchronization errors of full block sizes are recoverable.
- Some preprocessing is possible before a block is seen; the previous ciphertext block can be encrypted.
- Encryptions are not parallelizable; however, decryption is parallelizable and has a random-access property.

CFB is like CBC in that:

- Links plaintext together so the ciphertext depends on the preceding plaintext
- Input to the block algorithm needs to be kicked off with an IV, and the IV need not be secret
- In CFB, the IV needs to be unique, in CBC it should be but it isn't required

CFB (specifically 8-bit CFB) is the mode of choice for encrypting streams of characters when each character has to be treated individually, as in a link between a terminal and a host. However- for high-speed synchronous systems where error propagation is intolerable and preprocessing is required, OFB is the better option.

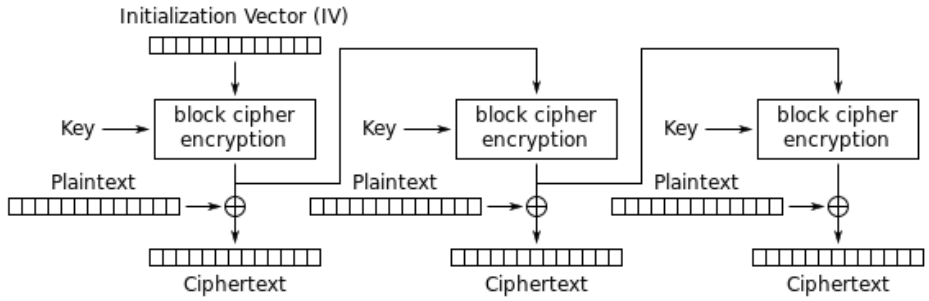
You can use CFB1-bit to do one bit at a time, but using one complete encryption with a block cipher on a single bit can be useless overhead. A stream cipher might be a better idea (plus, reducing the number of rounds to speed things up isn't recommended either!)

Plaintext is somewhat difficult for an attacker to manipulate; blocks can be removed from the beginning and end of the message, bits of the first block can be changed, and repetition allows some controlled changes.

OFB - Output Feedback mode

A block mode implemented as a synchronous stream cipher

The function of OFB is much like CFB, but instead of using the ciphertext block from the previously encrypted bits in it's shift register queue, it uses what the output function spits out *before* it gets XORed with the plaintext. It is sometimes called internal feedback since the feedback mechanism is *independent of the plaintext and ciphertext* (recall that in the definition of a synchronous stream cipher). Since it doesn't need the plaintext or ciphertext to do anything, most of the work can occur offline, before the plaintext message even exists. When the message finally arrives, it can be XORed with the output of the algorithm to produce the ciphertext.

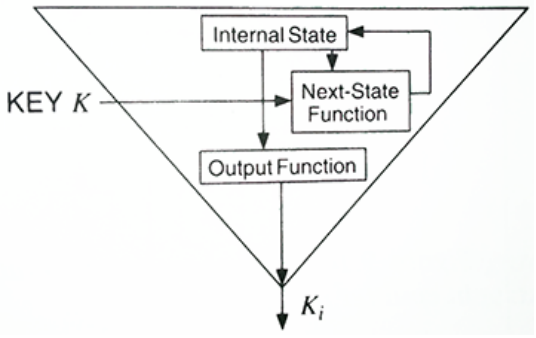


- Because it has no chaining dependencies, OFB doesn't suffer error propagation.
- A ciphertext error affects only the corresponding bit of plaintext
- Synchronization error is unrecoverable. Like with synchronized stream ciphers you need an external mechanism to detect synch loss and resych.

- Plaintext patterns are concealed. Input to the block cipher is randomized.
- Ciphertext is the same size as the plaintext, not counting the IV
- Processing is possible before the message is seen; is not parallelizable; counter processing is parallelizable.
- Plaintext is very easy to manipulate, any change in ciphertext directly affects the plaintext.
- The IV should be unique, doesn't need to be secret
- More than one message can be encrypted with the same key, provided that a different IV is used.

- It's recommended that OFB be used only when feedback size is same as block size (64-bit algorithm only in 64-bit OFB). Even though US Gov't authorizes other feedback sizes for DES, they should be avoided.
- OFB XORs a keystream with the text and the keystream will eventually repeat (the keystream's period).
- If it repeats using the same key then "there is no security"

- *"When the feedback size equals the block size, the block cipher acts as a permutation of m-bit values (where m is the block length) and the average cycle length is 2^m-1. For a 64-bit block length, this is a very long number. When the feedback size n is less than the block length, the average cycle length drops to around $2^{m/2}$. For a 64-bit block cipher, this is only 2^{32} - not long enough." - Schneier, Applied Cryptography*



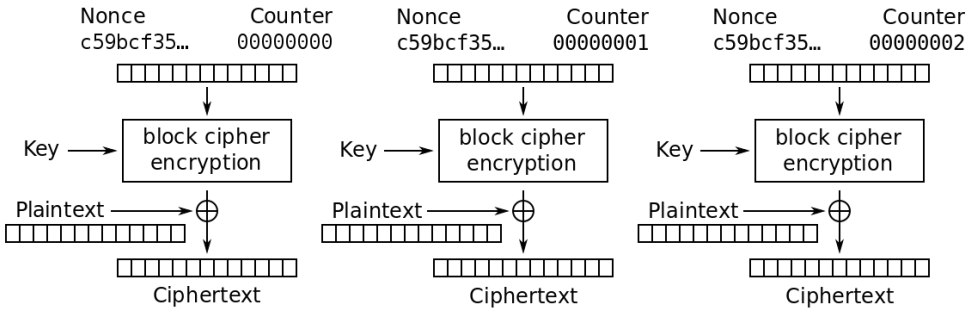
OFB operation as a stream cipher

OFB works slightly different when being used as a stream cipher First time in, internal state is given the IV (no previous key interaction) and is passed to the output function of the keystream generator without interacting with the key first. Simultaneously, that same content that was passed to the output function is incorporated with the key (in the next-state function) and placed in the new internal state (bumping the bits over like in CFB). Second time through the internal state contains the key-interacted previous internal state and part of the IV, and this is the first output containing key-involved stuff going to the output function.

$C_i = P_i \oplus S_i$; $S_i = E_K(S_i\text{-}1)$

Counter (CTR):
Like OFB, but uses a nonce and incrementing counter instead of an IV and encryption output to fill the register.

After each block encryption, the counter increments by some constant (often one, but designed to allow for less predictable options). Random-sequence generators can be used as input to the block algorithm, rather than more obvious sequences



Counter (CTR) mode encryption

- solves the OFB mode problem of n-bit output where n is less than the block length.
- synchronization and error propagation characteristics of this mode are identical to those of OFB.
- both encryption and decryption can be performed using many threads at the same time.

If one bit of a plaintext or ciphertext message is damaged, only one corresponding output bit is damaged as well. Thus, it is possible to use various correction algorithms to restore the previous value of damaged parts of received messages.

Sometimes called Segment Integer Counter (SIC) mode.

- This incrementing counter ensures each block is encrypted with a unique keystream (same content encrypts to different value), and provides the best performance.

Stream ciphers in counter mode have simple next-state functions and complicated output functions dependent on the key. The next-state function can be something as simple as a counter, adding one to the previous state.

With a counter mode stream cipher, it is possible to generate the ith key bit, ki, without first generating all the previous key bits. Simply set the counter manually to the ith internal state and generate the bit. This is useful to secure random-access data files; you can decrypt a specific block of data without decrypting the entire file.

As usual, a key should be changed after using it for encrypting an appropriate number of sent messages- but the CTR mode does allow for less frequent key changes. For example, AES in CTR should have a key change after about 264 plaintext blocks.

- 2001, NIST revised its list of approved modes by including AES as a block cipher and adding CTR mode in SP800-38A

- Jan 2010, NIST added XTS-AES in SP800-38E, CBC-CS/CTS in SP800-38A-Addendum " Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode

Confidentiality Modes Lack Authenticity, Integrity - Enter Message Authentication Code

- Traditional modes don't protect against accidental modification or malicious tampering,
- Detect with separate message authentication code such as CBC-MAC, or a digital signature.
- CBC-MAC is secure only for fixed-length messages
- Needed for dedicated integrity assurances, NIST approved HMAC, CMAC, and GMAC.
    - HMAC (Keyed-Hash MAC) was approved in 2002 as FIPS 198
    - CMAC (Cipher-based MAC) was released in 2005 under SP800-38B, aka OMAC (One-key MAC)
    - GMAC (Galois MAC) was formalized in 2007 under SP800-38D,

Compositing a confidentiality mode with an authenticity mode proved to be difficult and error prone
Solution: combine confidentiality and data integrity into a single cryptographic primitive.
The modes are referred to as authenticated encryption, AE or "authenc".

- Examples of AE modes are
    - CCM (Counter with CBC-MAC) - only defined for ciphers with a block length of 128 bits
    - https://en.wikipedia.org/wiki/CCM_mode
    - GCM (Galois/Counter Mode),
    - CWC,
    - EAX,
    - IAPM, and OCB.