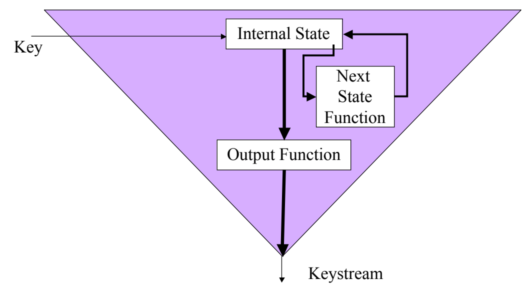
Keystream Generators
Random Number Generators
Block Ciphers
Chaining

1

Keystream Generator



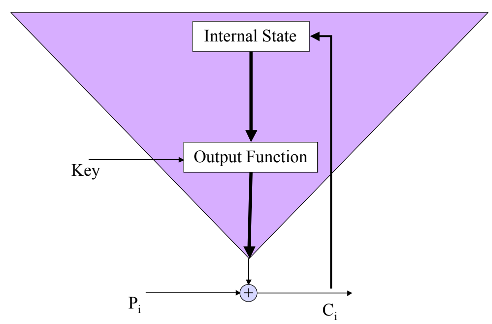
2

Synchronous Stream Cipher

- Keystream is generated from the key K
- Sender and receiver must be synchronized
- One-bit error in ciphertext produces one-bit error in plaintext
- Upon loss of synchronization both sides start afresh with a new key
- Any deletions and insertions will cause loss of synchronization
- Mallory can toggle/change bits

3

Self-Synchronizing Stream Cipher



4

Self-Synchronizing Stream Cipher

- Internal state is the function only of the previous *n* ciphertext bits and depends on the key K
- Decryption keystream generator will completely synchronize with encryption generator after receiving *n* bits
- Advantage:
  - Recovery from loss of bits after *n* bits
- Drawback:
  - Error extension – one-bit error in ciphertext produces *n* errors in plaintext
  - Mallory can replay messages

5

Generating Random Numbers

- We need to generate a sequence that looks random but is reproducible
- There shouldn't be any obvious regularities, otherwise Eve can learn the pattern after seeing several numbers, and guess the next ones
- We would like to cover the whole range of numbers (e.g. 2^n if the number has *n* bits)

6

Linear Congruential Generators

- Generators of the form

$$X_n = (aX_{n-1} + b) \bmod m$$

  - A period of a generator is number of steps before it repeats the sequence
  - If *a*, *b* and *m* are properly chosen, this generator will be *maximal period generator* and have period of *m*
  - It has been proven that any polynomial congruential generator can be broken

7

Linear Feedback Shift Registers

- Used for cryptography today
- A **shift register** is transformed in every step through **feedback function**
  - Contents are shifted one bit to the right, the bit that “falls out” is the output
  - New leftmost bit is XOR of some bits in the shift register, **tap sequence**
  - If we choose a proper tap sequence period will be 2^n-1

8

Linear Feedback Shift Registers

$$X^4 = X^4 \oplus X^1$$

1111		0110	1	1000	1
0111	1	0011	0	1100	0
1011	1	1001	1	1110	0
0101	1	0100	1	1111	0
1010	1	0010	0		
1101	0	0001	0		

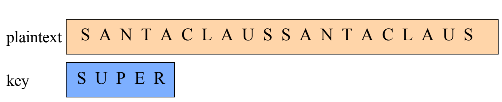
9

Linear Feedback Shift Registers

- Proper tap sequences are those where a polynomial from a tap sequence + 1 is a primitive polynomial in GF(2)
- There are tables of primitive polynomials (I posted some of them on our class page)
- LFSR is fast in hardware but slow in software
- LFSR are not themselves secure but they are used as building blocks in encryption algorithms

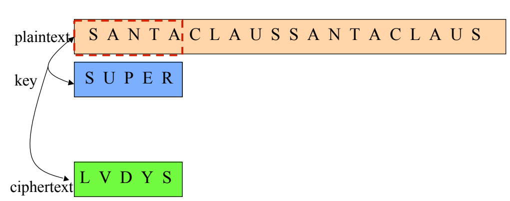
10

Block Cipher Example



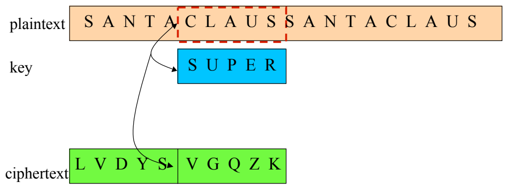
11

Block Cipher Example

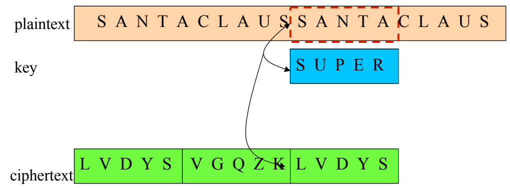


12

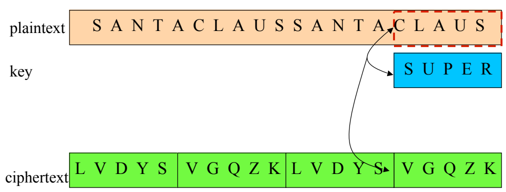
Block Cipher Example



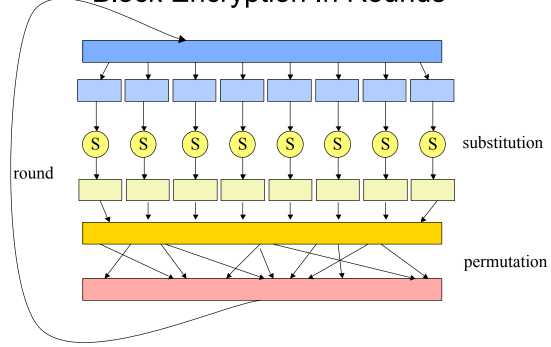
Block Cipher Example



Block Cipher Example



Block Encryption In Rounds



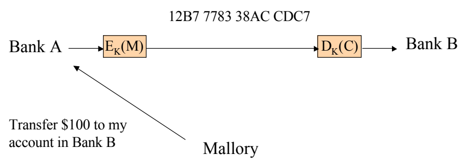
Encrypting A Large Message

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- *k*-bit Cipher Feedback Mode (CFB)
- *k*-bit Output Feedback Mode (OFB)
- Things to consider:
  - Can we encrypt/decrypt efficiently (as soon as bits arrive)
  - How hard it is to break encryption
  - What if a bit is flipped on the channel
  - What if we lose a bit on the channel

Electronic Code Book (ECB)

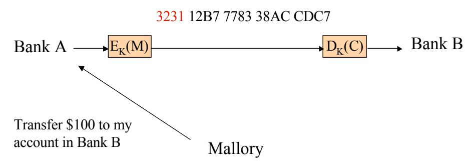
- Precompute and store mapping for every possible block
  - Fast encryption/decryption – just a table lookup
  - Ability to process text in any order and in parallel
  - Table size could be enormous even for 64 bit blocks so we need to make the mapping depend on the key
- Eve can detect which blocks map to other blocks, by seeing several plaintext and corresponding ciphertext messages
- Due to language redundancy even partial decryption might provide enough information
- Bit error invalidates one block
- Bit loss/addition is not recoverable

Block Replay



- Mallory does this couple of times, looks for similar block sequences.
- She can now replay 12B7 7783 38AC CDC7 at will

Block Replay

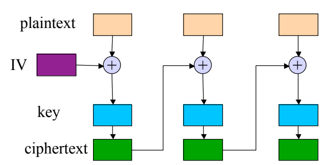


- Bank adds timestamps
- Mallory picks specific blocks of message carrying his name and account number and replaces those in other messages between Bank A and Bank B

Cipher Block Chaining (CBC)

- Problem with ECB is that Mallory can replace, add or drop blocks at will
- Chaining prevents this by adding feedback
  - Each ciphertext block depends on all previous blocks
- Also, with CBC, same plaintext blocks will encrypt to different ciphertext blocks thus obscuring patterns in plaintext

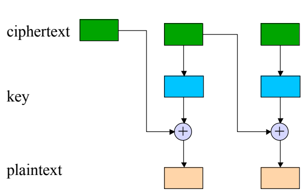
Cipher Block Chaining (CBC)



Encryption

Initialization vector (IV) is just a block of random numbers, to ensure that no messages have the same beginning
Both the sender and the receiver must use the same IV.

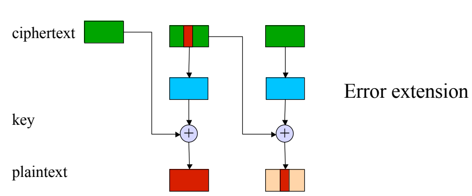
Cipher Block Chaining (CBC)



Decryption

Error Recovery

- An error in plaintext affects the rest of the message but is easily spotted and removed after decryption
- An error in ciphertext affects one block and several bits of plaintext



Potential Problems With CBC

- Mallory can:
  - Add blocks
  - Drop blocks
  - Introduce bit errors
- Bit loss/addition is not recoverable

25