Stream Cipher

CSCI 454/554

Advanced Block Ciphers

• features seen in modern block ciphers are:
  – variable key length / block size / number rounds
  – mixed operators, data/key dependent rotation
  – key dependent S-boxes
  – more complex key scheduling
  – operation of full data in each round
  – varying non-linear functions
Stream Ciphers

- process the message bit by bit (as a stream)
- typically have a (pseudo) random stream key
- combined (XOR) with plaintext bit by bit
- randomness of stream key completely destroys any statistically properties in the message
  \[ C_i = M_i \text{ XOR StreamKey}_i \]
- what could be simpler!!!!
- but must never reuse stream key
  – otherwise can remove effect and recover messages

\[
\begin{align*}
\text{Encryption} & : \\
\text{Key} & \rightarrow \text{Pseudorandom byte generator} \\
\text{M} & \rightarrow k \\
& \rightarrow \text{Ciphertext byte stream C} \\
\text{Decryption} & : \\
\text{Key} & \rightarrow \text{Pseudorandom byte generator} \\
\text{C} & \rightarrow k \\
& \rightarrow \text{Plaintext byte stream M}
\end{align*}
\]
Stream Cipher Properties

- some design considerations are:
  - long period with no repetitions
  - statistically random
  - depends on large enough key
  - large linear complexity
  - correlation immunity
  - confusion
  - diffusion
  - use of highly non-linear boolean functions

RC4

- a proprietary cipher owned by RSA DSI
- another Ron Rivest design, simple but effective
- variable key size, byte-oriented stream cipher
- widely used (web SSL/TLS, wireless WEP)
- key forms random permutation of all 8-bit values
- uses that permutation to scramble input info processed a byte at a time
RC4 Key Schedule (Initialization)

♦ starts with an array $S$ of numbers: $0..255$
♦ use key to well and truly shuffle
♦ $S$ forms **internal state** of the cipher
♦ given a key $k$ of length $L$ bytes
  
  ```
  for $i = 0$ to $255$ do
    $S[i] = i$
    $T[i] = K[i \mod L]$
  
  $j = 0$
  for $i = 0$ to $255$ do
    $j = (j + S[i] + T[i]) \mod 256$
    swap $(S[i], S[j])$
  ```

Stream Generation/Encryption

♦ encryption continues shuffling array values
♦ sum of shuffled pair selects "stream key" value
♦ XOR with next byte of message to en/decrypt

  ```
  i = j = 0
  for each message byte $M_i$
    $i = (i + 1) \mod 256$
    $j = (j + S[i]) \mod 256$
    swap$(S[i], S[j])$
    $t = (S[i] + S[j]) \mod 256$
    $C_i = M_i \text{ XOR } S[t]$
  ```
RC4 Security

- claimed secure against known attacks
  - have some analyses, none practical
- result is very non-linear
- since RC4 is a stream cipher, must **never reuse a key**
- have a concern with WEP, but due to key handling rather than RC4 itself