Travis Collins
ECE578 Lecture 3 Notes
September 17, 2012

RC4 $Ron'sCipher4$

-Designed in 1987
-Trade Secret of RSA Corp
-Leaked on sci.crypt $USENET$ in 1994
-Most widely used stream cipher, SSL/TLS, WEP/WPA
-Key Advantage: Amazingly simple/easy to implement!!
-RC4 works with bytes (8-bits) and not bits
-RC4 State

1. A 256 byte state table
2. Two 8-bit indices i, j

RC4 Key Schedule Algorthim

-Prepares the state table S using a short key or password
-Key has to be at least 1 byte

$$1 <= |key| <= 256 bytes, key = n$$

Algorithm

```
for i=0 to 255: S[i]=i
j=0
for i=0 to 255:
j=(j+S[i]+key[i mod n])mode 256
swap(S[i],S[j])
```

Key Stream Generation Algorithm

-In each iteration we generate a byte of keystream data
-Initially set i=j=0 (Only at beginning of encryption session)

Algorithm

```
i=i+1 (mod 256)
j=j+S[i] (mod 256)
swap(S[i],S[j])
return S[ S[i]+S[j] (mod 256)]
```

Block Ciphers -Remember (from stream ciphers) PRNG output string is indistinguishable from a random string for any bounded adversary

Idea: What if we can randomize the function itself instead of the output of the function!
-Pseudo Random Function (PRF)

Def: A PRF is a keyed function that is indistinguishable from a function chosen at random using bounded resources

Block Cipher(Approx. of Pseudo-Random Permutation (PRP))
is stateless meaning the same message and key in means the same cipher text out

$$E_k(m) = cD_k(E_k(m)) = m)$$

An Ideal Block Cipher -Assume we fix

$$n_k = n_m = n_c = n$$

-What we need is a random function n-bit to n-bit function
-Consider first all functions

Message	Cipher
0	2^n
1	2^n
...	...
$2^n - 1$	2^n

$$|F| = (2^n)^{2^n}$$

-But we want decryption to work so f needs to be one-to-one

Message	Cipher
0	2^n
1	$2^n - 1$
...	...
$2^n - 1$	1

$$|F| = 2^n(2^n - 1)(2^n - 2)... = (2^n)!$$

Still huge space!!
But we also want it to be efficiently computable

Can we construct a PRP from a PRF?
- Luby-Rackoff Construction-(Feistel Cipher)
- Look at DES paper