Stream ciphers

Replace the random key by a pseudorandom key

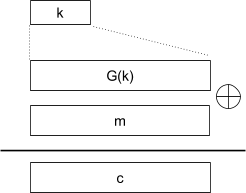
PRG:

Function

Efficient

Deterministic

Output must look random



# Examples of Real-World Stream Cyphers

## RC4 (1987)

Takes an initial seed (128bits for example) and expands it to a 2048 bit pseudo-random stream. Then initiates a loop where every iteraction of this loop outputs a byte from these 2048 bits.

Used in HTTPS and WEP

Weakness:

1.- Bias in initial output:

2.- Prob of (0,0) is 1/2562 + 1/2563

3.- Related key attacks

## CSS (Content Scrambling System)

Used on DVDs

Linear feedback shift register (LFSR)

DVD encryption (CSS): 2 LFSR

GSM encryption (A5/1,2): 3 LFSR

Bluetooth (E0): 4 LFSR

## CSS

seed = 5 bytes = 40 bits

1 LSFR: 17 bits <- 1 || first 2 bytes of key

2 LSFR: 25 bits <- 1 || last 3 bytes of key

Generates 8 bits of output from each LFSR. sums both bytes (plus carry from prev block), mod 256.

## Modern stream ciphers

### sStream

PRG: {0,1}s x R -> {0,1}n

Nonce: a non-repeating value for a given key

E(k,m;r) = m (xor) PRG(k;r)

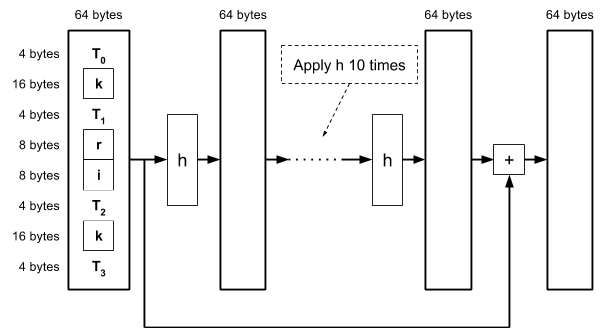
The pair (k,r) is never userd more than once. We can reuse the key because the pair (k,r) is unique.

### eStream: Salsa 20 (SW and HW)

Salsa20: {0,1}128 or 256 x {0,1}64 -> {0,1}n max n = 273 bits

Salsa20(k;r) := H(k,(r,0)) || H(k,(r,1)) || H(k,(r,2)) || …

Prepares a 64 bit block formed by:



h is an invertible function designed to be fast on x86 (SSE2)

#### Performance

RC4

Salsa20/12

Sosemanuk

## PRG Security Definitions

Let G:K -> {0,1}n be a PRG

Goal: define what it means that an output of the PRG is “indistinguishable” from a random output

### Statistical tests

A statistical test on {0,1}n: is an algorithm A such that A(x) outputs “0” or “1”

0: not random

1: random

### Advantage

Let G:K -> {0,1}n be a PRG and A a statistical test on {0,1}n

Define Advantage:

AdvPRG[A,G] := | Pr[A(G(k))=1] - Pr[A(r)=1] |

### Secure PRG

We say that G:K -> {0,1}n is a secure PRG if for all efficient statistical tests A, AdvPRG[A,G] is “negligible”

A secure PRG is unpredictable

An unpredictable PRG is secure

if PRG G is unpredictable at position i the G is a secure PRG

### Computationally indistinguishable

Let P1 and P2 be two distributions over {0,1}n

P1 and P2 are computationally indistinguishable if for all efficient statistical tests A

| Prx<-P1[A(x)=1] - Prx<-P2[A(x)=1] | < negligible

a PRG is secure if computationally indistinguishable from a uniform distribution

## Semantic Security