09-Randomness
TOUR OF ACCOUNTING

OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.

NINE NINE NINE NINE NINE

ARE YOU SURE THAT’S RANDOM?

THAT’S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.
```c
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
    // guaranteed to be random.
}
```
RNG_CreateContext() {

    (seconds, microseconds) = time of day; /* Time elapsed since 1970 */

    pid = process ID; ppid = parent process ID;

    a = mklcpr(microseconds);

    b = mklcpr(pid + seconds + (ppid << 12));

    seed = MD5(a, b); /* seed is a global variable */
}

mklcpr(x) { /* not cryptographically significant; shown for completeness */
    return ((0xDEECE66D * x + 0x2BBB62DC) >> 1);
}

[Randomness and the Netscape Browser] Dr. Dobb's, January 1996.
Netscape 1.1 Seeding Process

RNG_GenerateRandomBytes() {
    x = MD5(seed);
    seed = seed + 1;
    return x;
}

global variable challenge, secret_key;
create_key() {
    RNG_CreateContext();
    tmp = RNG_GenerateRandomBytes();
    tmp = RNG_GenerateRandomBytes();
    challenge = RNG_GenerateRandomBytes();
    secret_key = RNG_GenerateRandomBytes();
}

GW [Randomness and the Netscape Browser] Dr. Dobb's, January 1996.
Jone’s RNG Rules

1. Don’t use system generators
2. Use a known good RNG you implemented
3. Properly seed the RNG
KISS Generator (G. Marsaglia)

```c
static unsigned int /* Seed variables */
    x = 123456789,
    y = 362436000,
    z = 521288629,
    c = 7654321;

unsigned int KISS()
{
    unsigned long long t, a = 698769069ULL;
    x = 69069*x+12345;
    // y never == 0! */
    y ^= (y<<13); y ^= (y>>17); y ^= (y<<5);
    t = a*z+c; c = (t>>32); // Also avoid setting z=c=0!
    return x+y+(z=t);
}
```