

MIT OpenCourseWare
<http://ocw.mit.edu>

6.033 Computer System Engineering
Spring 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

6.033 Lecture 22

Nickolai Zeldovich

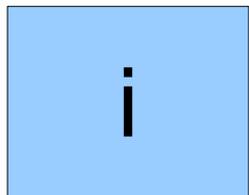
RC4

Initialization

$S[0..255] =$
permutation of $0..255$
(based on key)

$i = 0$

$j = 0$



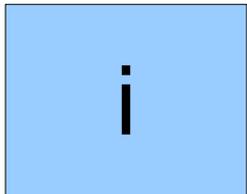
RC4

Initialization

```
S[0..255] =  
  permutation of 0..255  
  (based on key)  
i = 0  
j = 0
```

Generate pseudo-random byte

```
i = (i + 1) mod 256  
j = (j + s[i]) mod 256  
swap S[i] ↔ S[j]  
x = (S[i] + S[j]) mod 256  
return S[x]
```



RSA

Initialization

$p = \text{random large prime}$

$q = \text{random large prime}$

$N = p * q$

A blue square with a black border, containing the letter 'N' in the center.

RSA

Initialization

$p = \text{random large prime}$

$q = \text{random large prime}$

$N = p * q$

$e = \text{random, does not}$
 $\text{divide } (p-1)*(q-1)$

Compute d , such that
 $d * e \equiv 1 \pmod{(p-1)*(q-1)}$

e

N

d

RSA

Initialization

p = random large prime
 q = random large prime

$N = p * q$
 e = random, does not
divide $(p-1)*(q-1)$
Compute d , such that
 $d * e == 1 \text{ mod } (p-1)*(q-1)$

Encrypt(m, N, e) $\rightarrow c$:

$$c = m^e \text{ mod } N$$

Decrypt(c, N, d) $\rightarrow m$:

$$m = c^d \text{ mod } N$$

