6.033 Computer System Engineering

Spring 2009

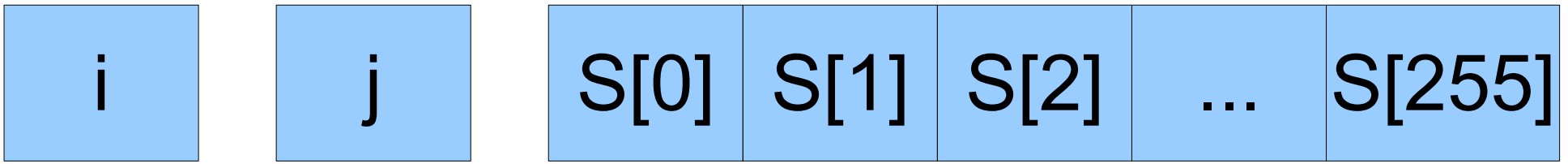
6.033 Lecture 22

Nickolai Zeldovich

RC4

```
Initialization

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



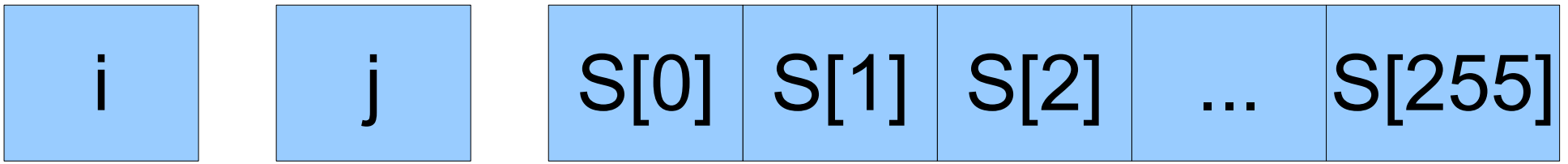
RC4

```
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]
```

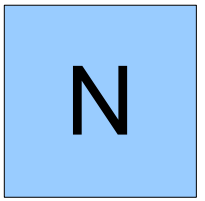


RSA

```
Initialization

p = random large prime
q = random large prime

N = p * q
```

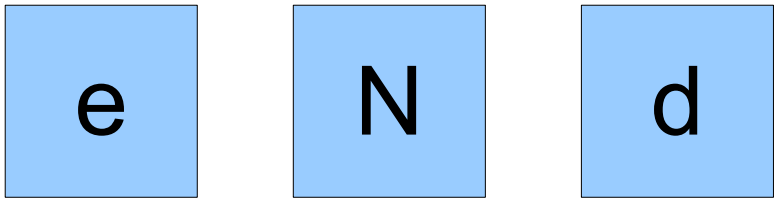


RSA

```
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 mod (p-1)*(q-1)
```



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 mod (p-1)*(q-1)

Encrypt(m, N, e) → c:

$c = m^e \bmod N$

Decrypt(c, N, d) → m:

$m = c^d \bmod N$