CSE 321 Discrete Structures

February 10th, 2010

Lecture 15: Public Cryptography

Announcements

- Midterms are graded (see catalysttools)

- Homework is due on Friday as usual

- No class on Monday

Cryptography

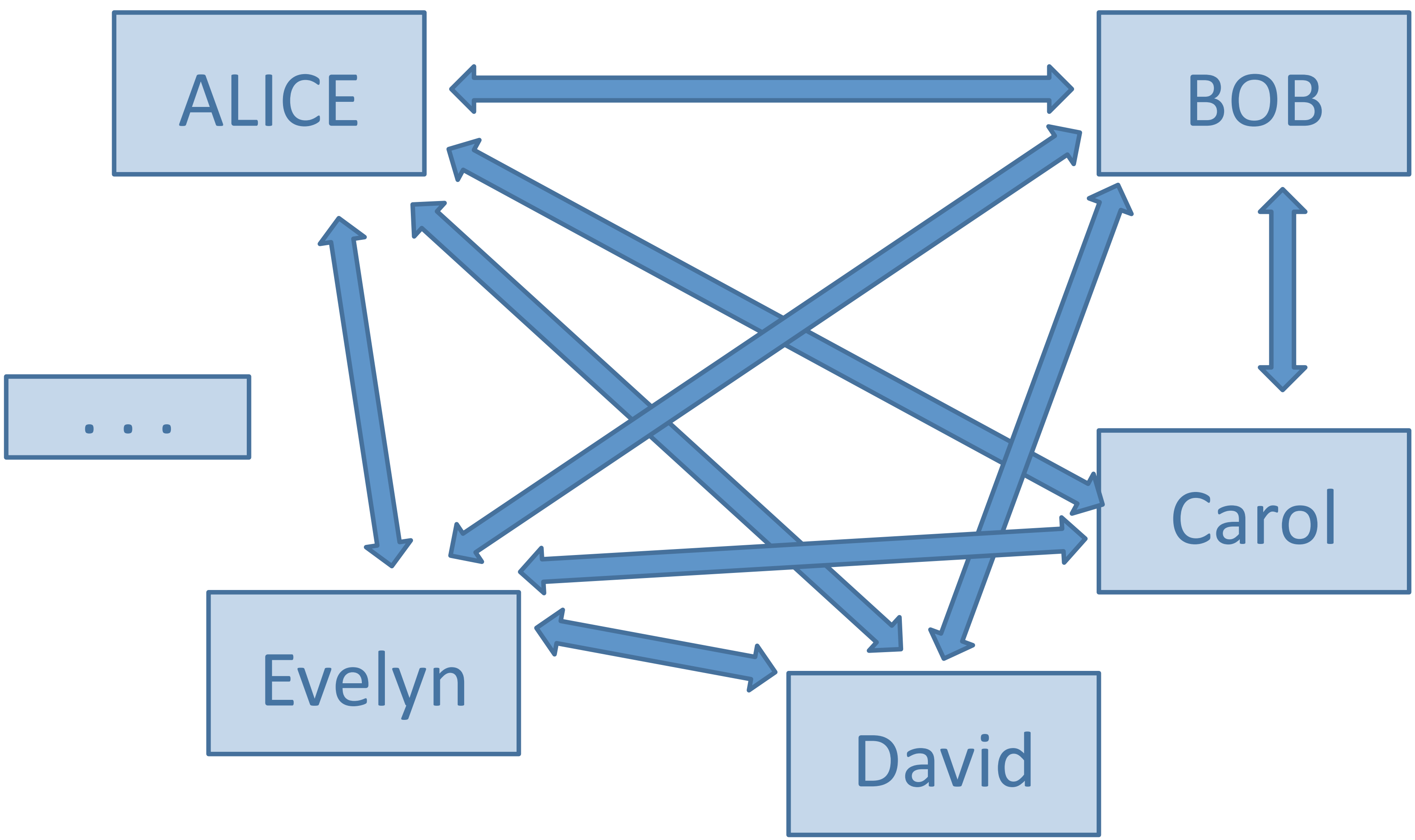
Perfect encryption

"One time pad":

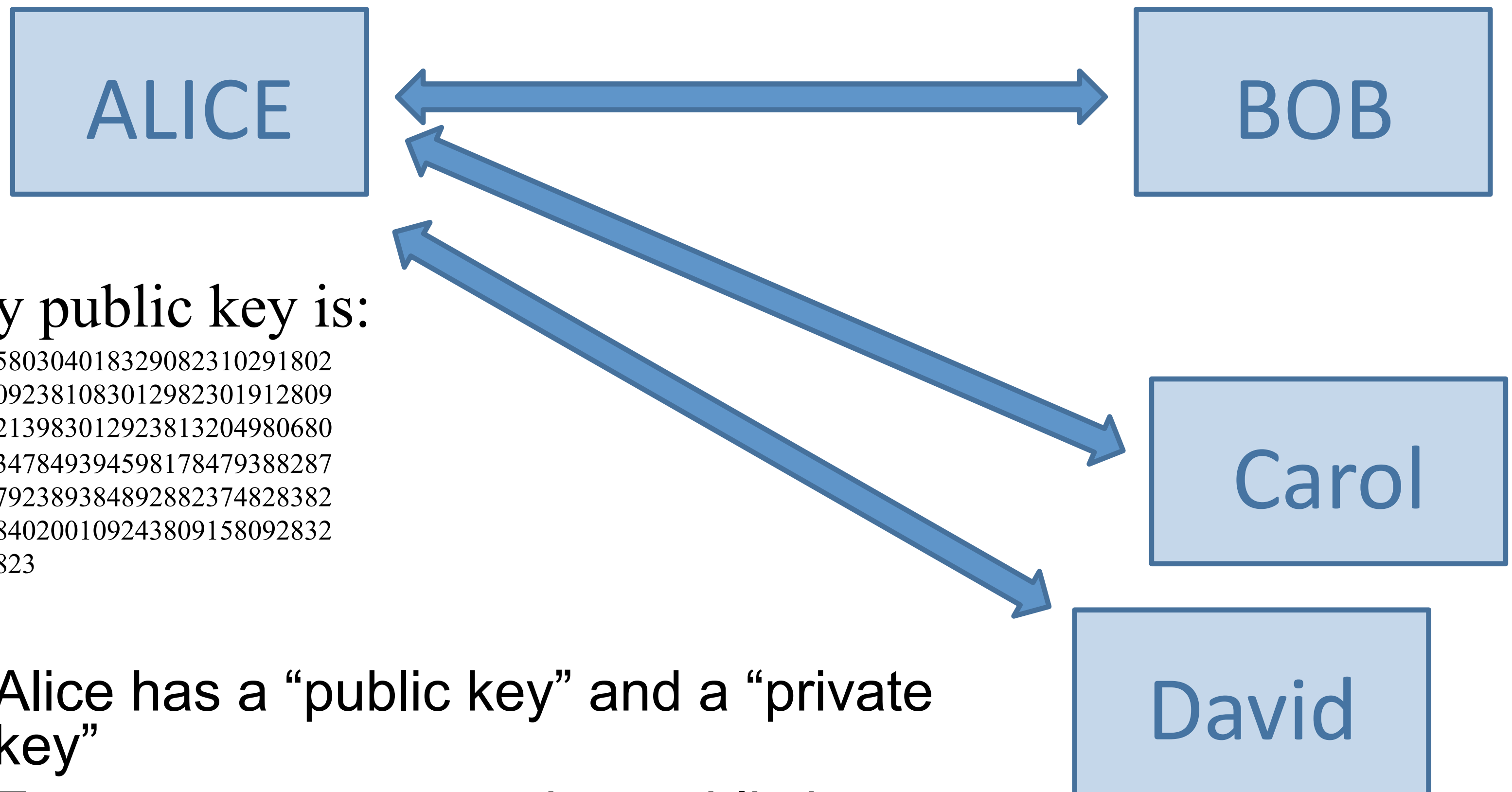
- Alice and Bob have a shared n-bit secret S
- To send an n-bit message M, Alice sends M $\oplus$ S to Bob
- Bob receives the message N, to decode, Bob computes N $\oplus$ S
- Note: can't reuse S (various attacks exists)
- But if S is unique, then this perfect encryption

What is the problem with the one time pad ?

The Key Distribution Problem



Public Key Cryptography



My public key is:
1389058030401832908231029180
2198210923810830129823019128
0921830213983012923813204980
6802980934784939459817847938
8287398457923893848928823748
2838299293840200109243809158
09283290823823

- Alice has a "public key" and a "private key"
- Everyone encrypts w. her public key
- She decrypts with her private key

RSA

- Rivest – Shamir – Adelman

1. Choose p, q are large primes
2. Let n = pq.
3. Choose e relatively prime to (p-1)(q-1)
4. Let d = e^{-1} mod (p-1)(q-1)
5. Alice publishes:
   1. n
   2. e as the encryption key
6. Alice keeps private:
   1. p and q
   2. d as the decryption key

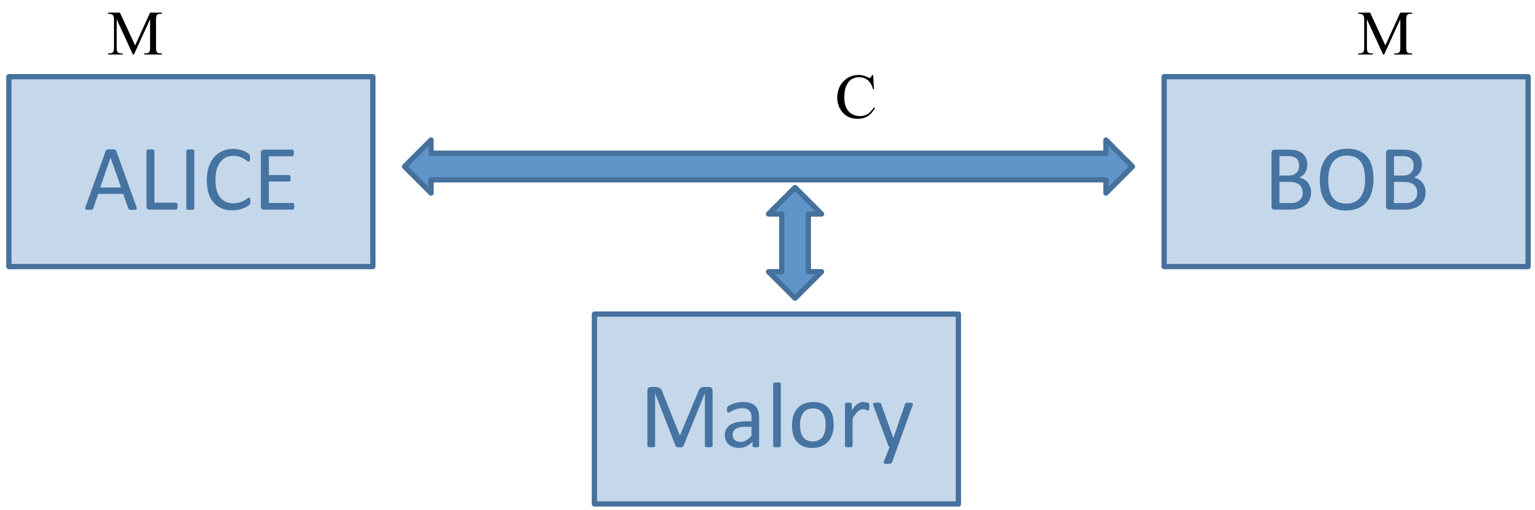
Message protocol

- Bob
  - Read e, n from Alice's public site
  - To encrypt message M: $C = M^e \bmod n$
  - Send C to Alice
- Alice
  - Receive C from Bob
  - To decrypt message C: $M = C^d \bmod n$

Why Decryption Works

- $d = e^{-1} \bmod (p-1)(q-1) \rightarrow de = 1 + k(p-1)(q-1)$
- $C^d \equiv (M^e)^d = M^{de} = M^{1 + k(p-1)(q-1)} \pmod{n}$
- $C^d \equiv M\,(M^{p-1})^{k(q-1)} \equiv M \pmod{p}$
- $C^d \equiv M\,(M^{q-1})^{k(p-1)} \equiv M \pmod{q}$
- Hence $C^d \equiv M \pmod{pq}$
  - By the Chinese remainder theorem
- Note: we must have $M < p$ and $M < q$

Why RSA is Secure



- Malory (the bad guy) wants to steal M
- Malory knows: C, d, n
- Malory needs: d (= e^{-1} mod (p-1)(q-1))
- Nobody knows how to compute e better than this:
  - Factor n = p*q
  - Compute d = e^{-1} mod (p-1)(q-1)
- Nobody knows how to factor n efficiently

Practical Aspects

How do we find a large primes p, q ?

- Choose a number p randomly
- Test if it is prime
  - How ??? Next lecture
- If not prime, choose another number
  - How many times do we need to try ??? Next lecture

Practical Aspects

- How do we compute $d = e^{-1} \bmod (p-1)(q-1)$?
  - Note that $GCD(e, (p-1)(q-1)) = 1$
  - Extended Euclid's algorithm: find d, k s.t.: $d*e + k*(p-1)*(q-1) = 1$
- How do we exponentiate efficiently ?
  - Bob: $M^e \bmod n$
  - Alice: $C^d \bmod n$
  - Use "fast exponentiation"
    - Next lecture, but recall "fast multiplication" !