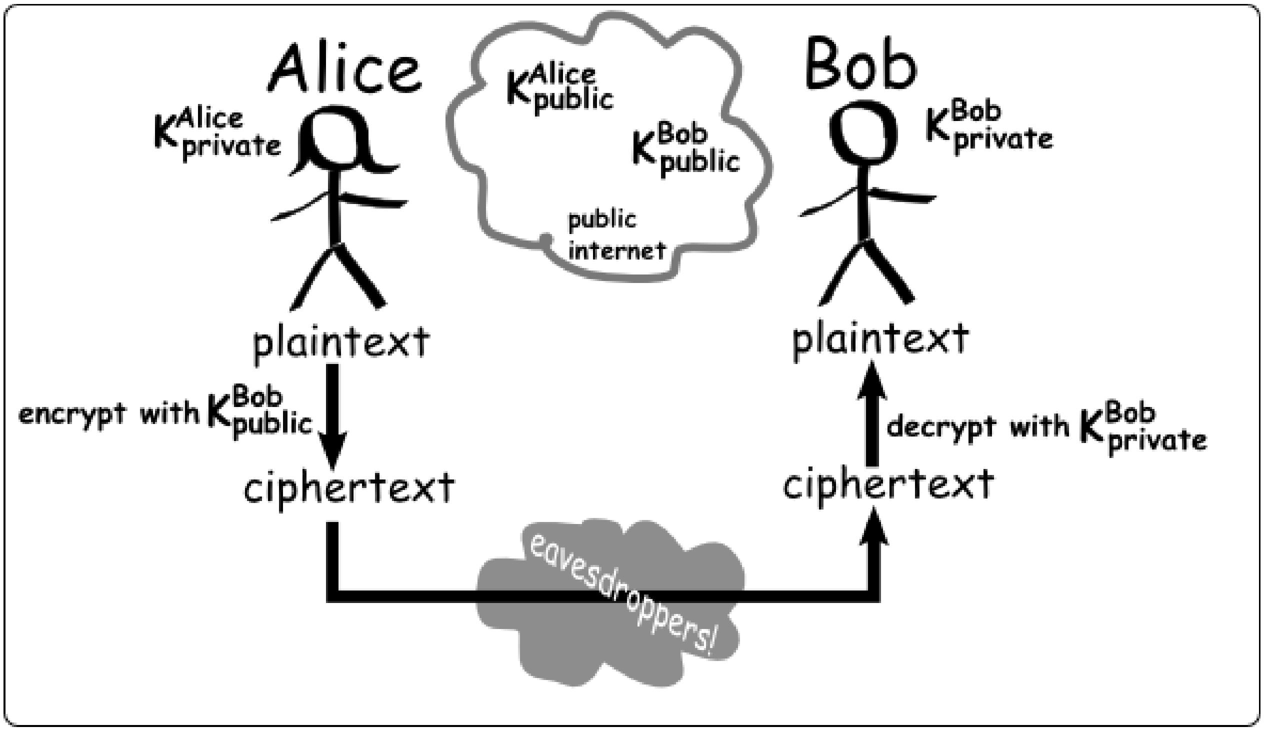
Public Key Cryptography (PKC)

general idea

- uses two keys: one for encryption; the other for decryption
- it is not possible to use the same key to both encrypt and decrypt the same message

working mechanism

- if *E* and *D* are encryption and decryption functions respectively, and *k1* and *k2* are the set of keys, then:
  - $E_{k1}(P)=C$ or $E_{k2}(P)=C$
  - $D_{k2}(C)=P$ $D_{k1}(C)=P$



notes about PKC

- the private key must be kept secret
- it is infeasible to compute an unknown key from the known key
- other users in the network wanting to communicate only have access to the recipient’s public key
- a message encrypted using a public key can be decrypted only using its corresponding private key
  - (and vice-versa – for digital signatures)

PKC specifics

- we'll discuss more specific PKC algorithms later
- but for now, most PKC algorithms use very large numbers (with hundreds of digits)
  - the public key is the large number
  - the private key is one of two prime factors of the large number
  - it is very hard to factor a large number as the product of two prime numbers
- e.g.:

21	=	3	*	?
589	=	31	*	?
8,633	=	97	*	?

- e.g. (a 100-digit number – which is “small”):
    - 1,522,605,027,922,533,360,535,618,378,132,637,429,718,068,114,961,380,688,657,908,494, 580,122,963,258,952,897,654,000,350,692,006,139
    - (1.5 duotrigintillion)
    - =
    - 37,975,227,936,943,673,922,808,872,755,445,627,854,565,536,638,199
    - (37.9 quindecillion)
    - *
    - 40,094,690,950,920,881,030,683,735,292,761,468,389,214,899,724,061
    - (40.1 quindecillion)

- e.g. (RSA-1024: 1,024 bits = 309 digits – which is still “small”):
    - 135,066,410,865,995,223,349,603,216,278,805,969,938,881,475,605,667,027,524,485,143,851, 526,510,604,859,533,833,940,287,150,571,909,441,798,207,282,164,471,551,373,680,419,703, 964,191,743,046,496,589,274,256,239,341,020,864,383,202,110,372,958,725,762,358,509,643, 110,564,073,501,508,187,510,676,594,629,205,563,685,529,475,213,500,852,879,416,377,328, 533,906,109,750,544,334,999,811,150,056,977,236,890,927,563
    - (135 cenuntillion)

    - think about having to factor this into the product of two primes (even computationally)
        - it's basically intractable!

issues with PKC

- it's computationally more complex than symmetric cryptography
- because of this, it is usually only used to encrypt a small amount of data
- it is prone to a chosen-plaintext attack for very short messages (if the ciphertext is known)
- some forms of man-in-the-middle attacks are effective against the cryptosystem that uses PKC

hybrid cryptosystems

- when more than one form of cryptography is used within the same cryptosystem
- most secure cryptosystems use PKC only for securing a key used in symmetric cryptography
    - the actual data exchange is done using the key that was transmitted using PKC
    - this is due to the computational complexity involved with PKC
- some symmetric cryptography algorithms provide as much security as PKC
    - if the shared key is kept truly secret

the concept of PKC plays a large role in the implementation of digital signatures

more on PKC later!