## Public Key Cryptography (vs Symmetric)

* Symmetric key cryptography
  + Requires sender, receiver know shared secret key (so requires key distribution)
  + Q: How do you even agree on the key in the first place (especially if the parties have never “met”)? A: Use public key cryptography!
* Public-key cryptography
  + Radically different approach (Diffie-Hellman76, RSA78)
  + Sender, receiver do NOT share secret key
  + Public encryption key known to ALL
  + PRIVATE decryption known only to receiver

## Public key encryption

Timeline

Description automatically generated with medium confidence

* Requirements:
  + Need and such that:
  + Given public key , it should be impossible to compute private key
  + Use RSA (Rivest, Shamir, Adelman) encryption algorithm
* How the keys are like:
  + **Key pairs:** unlike symmetric crypto, this requires a pair of keys
  + **Public key:** by nature, designed to be “public”; can be freely given to anyone
  + **Private key:** must be kept confidential and never shared
  + **Both directions**: keys can work in both directions

## RSA: Choosing keys

1. Choose two large prime numbers , such that their product has at least 1024 bits
2. Compute
3. Choose (with ) such that are relatively prime.
4. Choose such that is exactly divisible by . (in other words: )  
   % = modulus operator.
5. Public key is . Private key is .

## RSA: Encryption, decryption

Text, letter

Description automatically generated

## RSA: Why is that ^?

A picture containing table

Description automatically generated

## RSA Example:

A picture containing Word

Description automatically generated

Text

Description automatically generated

## Diffie-Hellman Key Exchange

Text, letter

Description automatically generated