Cryptography

Cryptography systems allow 2 parties to communicate securely. The intent is to give privacy, integrity and security to the information we store or transfer

What are the implications of this?

Cryptography is much more than…

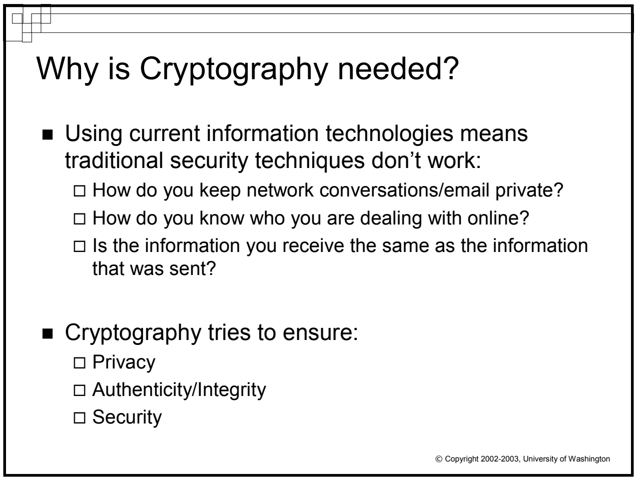
- Straight encoding and decoding
  - □ Usually a one for one representation of one character or datum for another
  - □ Morse Code
  - □ ASCII conversion
- Common characteristics of normal encoding
  - □ No secret formula used to convert data
  - □ Just a straight forward processing of data

What is Cryptography, exactly?

- "The art or science of keeping messages secure [using mathematics]."
  -*Applied Cryptography*
- Cryptography is the study of encryption and decryption methods
  - □ These methods usually involve very intense, high level math
- Cryptography relies on keeping some piece of the information (the key) secret

Why Cryptography? (cont'd)

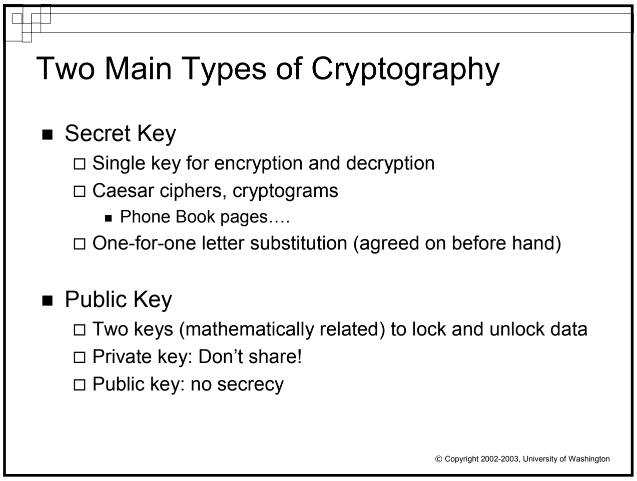
- Can be applied to any kind of electronic data:
  - □ Text
  - □ Audio
  - □ Video
  - □ Images
- Can be used real-time or for storage of data



Why is Cryptography needed?

- Using current information technologies means traditional security techniques don’t work:
  - How do you keep network conversations/email private?
  - How do you know who you are dealing with online?
  - Is the information you receive the same as the information that was sent?

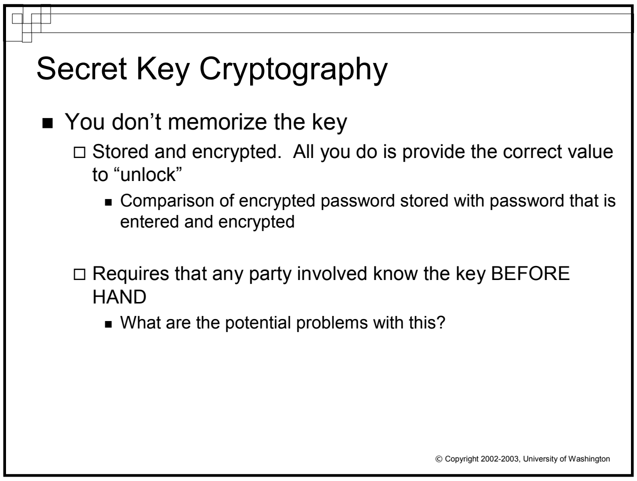
- Cryptography tries to ensure:
  - Privacy
  - Authenticity/Integrity
  - Security



Two Main Types of Cryptography

- Secret Key
  - Single key for encryption and decryption
  - Caesar ciphers, cryptograms
    - Phone Book pages….
  - One-for-one letter substitution (agreed on before hand)

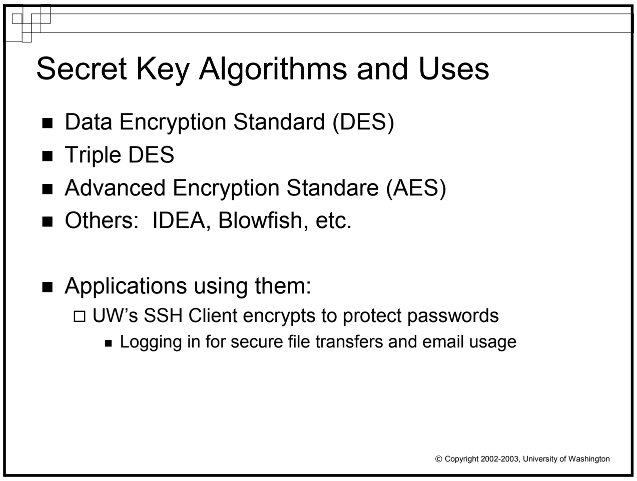
- Public Key
  - Two keys (mathematically related) to lock and unlock data
  - Private key: Don’t share!
  - Public key: no secrecy



Secret Key Cryptography

- You don’t memorize the key
  - Stored and encrypted. All you do is provide the correct value to “unlock”
    - Comparison of encrypted password stored with password that is entered and encrypted

  - Requires that any party involved know the key BEFORE HAND
    - What are the potential problems with this?



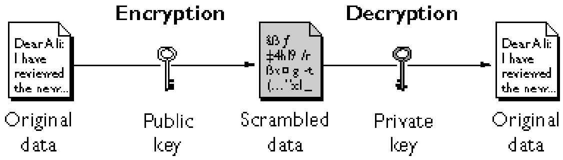
Secret Key Algorithms and Uses

- Data Encryption Standard (DES)
- Triple DES
- Advanced Encryption Standare (AES)
- Others: IDEA, Blowfish, etc.

- Applications using them:
  - UW’s SSH Client encrypts to protect passwords
    - Logging in for secure file transfers and email usage

Public Key Cryptography

- Most famous algorithm: RSA
  - □ Named after its creators: Rivest, Shamir and Adleman
- Critical that sender and receiver have a common key, the public key
- Security relies on difficulty of finding factors of very large numbers



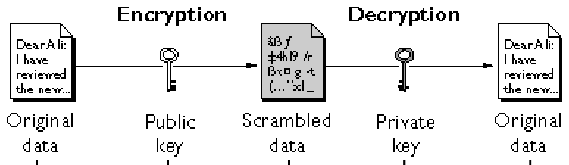
How does RSA work?

- Receiver set up:
  - □ Choose a couple large prime numbers (200 digits or more), p and q (make sure both are 2 larger than a multiple of 3)
  - □ Multiply p and q to get n
n=p*q
  - □ Receiver also computes s
s=(1/3)(2(p-1)(q-1) +1)
  - □ n is your public key: publish it
    - Keep p, q and s private

How does RSA work? (cont'd)

- Sender obtains public key (n) and encrypts message:
  - □ Convert message into chunks (multiple byte chunks)
  - □ Translate each chunk into an integer, m
  - □ Now, it gets a little tricky…..

    Divide m^3 by n and the remainder is your encrypted text, call it E $E= m^3/n$
- Receiver decrypts message:
  - □ Divide E^S by n and the remainder is your clear text, or original message integer, m which can now be converted back to the appropriate letter: $C= E^S /n$
    - Remember, s was not given out and is only known by the receiver!



Sender wants to send information

Sender gets public key, chunks data, converts data to bits, each chunk is cubed, divided by n and the remainder sent as cipher text: $E = m^3/n$

Chunks are sent to receiver

Receiver puts cipher text chunks to the s power, divides by n and the remainder is the original clear text chunk (in bits), convert back to text or audio or color, etc. $C = E^S /n$

Receiver puts chunks together to get original data

Why is RSA secure?

- If you know n, then you can get p and q and therefore s, right……
  - □ Well, sort of….Remember:
    -p and q are VERY large
    -n is even larger (p*q)
  - □ To find s, you need p and q, but all you have is a VERY LARGE n. You need to factor n to find both p and q
  - □ Factoring a number means representing it as the product of prime numbers
    - Easy for computers to do UP TO A POINT
    - Very Large numbers: more computer time needed than all the life times of you through your great-grandchildren's, great-grandchildren.

Some Public Key Algorithms and Uses

- RAS (Rivest, Shamir, Adelman)
- DSA (Digital Signature Algorithm)
- Applications using them:
  - □ Email
  - □ Financial Transactions
  - □ Browsers
  - □ Mobile Telecommunications
  - □ E-voting
  - □ DVD encryption
  - □ ….

Unbreakable code: Pros and Cons

- So, if crypto systems using algorithms like RSA and others are now virtually unbreakable…..
  - □ Do we have total security?
  - □ Privacy?
  - □ Integrity?
  - □ For WHO?
    - When is the unbreakable code good? Bad?

Summary

- Cryptography is one way to provide security services

- Two main types
  - □ Secret-key crypto: Mainly for encryption/decryption where key is agreed upon prior, or encryption is one-way
  - □ Public-key crypto: Publish public key, receiver keeps private key