ISA 563: Fundamentals of Systems Programming

Hashing and Encryption

April 16, 2012

Hashing

- Hashing is a deterministic mapping from variable-sized input data to a fixed sized output
  - a procedure that performs such a mapping is called a hash function
- Since input size is infinite, and output size is finite:
  - Output is only a digest (checksum) of input, and input cannot (normally) be restored from output
  - there will be collisions:
    - A collision is a scenario where to different inputs hash to the same output

Hashing Functions

- Ideal properties of hash functions:
  - should be easy to compute
  - should make collision infeasible:
    - should be hard to find another message the same hash function
    - should be hard to alter current message without changing the hash
    - should be hard to find two different messages with the same hash

Hash Functions (cont'd)

- What do hash functions give us?
  - Just as many other fucntions, hash functions has a many to one mapping from input to output
  - Scenario: we apply the same hash function to two sets of input data:
    - if the two hash values are different, then the two input data sets are different
    - if the two hash values are the same, they the inputs are probably the same
      - degree of certainty depends on the hash function used
      - although can never be sure, should be enough for “practical” purposes

Demo

md5sum, sha1sum

Encryption

- Encryption is the transformation of an input to an output that can only be read by those who possess the correct key.
- Terminology:
  - plaintext: original input to be encrypted
  - ciphertext: encrypted output
  - key: a piece of data used for encryption/decryption
  - cipher: algorithm used to encrypt data

Encryption (cont'd)

- Simple model:
  - Encryption:
    - Enc = E(input, k1)
  - Decryption:
    - Dec = D(input, k2)
- Types:
  - Symmetric encryption:
    - k1 = k2
  - Assymetric encryption:
    - k1 != k2

Symmetric vs Assymetric Encryption

- Symmetric:
  - Both parties use the same key
    - key is shared
  - Faster encryption / decryption
  - Examples: DES, Triple-DES, IDEA, TWOFISH, BLOWFISH
- Assymmetric:
  - Use a pair of keys:
    - private key: kept secret by the owner
    - public key: published to everyone
  - Much slower than symmetric encryption
  - Examples: RSA, DSA, ELGAMAL

mcrypt library

- libmcrypt is a library that provides a uniform interface to several symmetric encryption algorithms:
    - DES
    - 3DES
    - RIJNDEAL
    - Twofish
    - IDEA
    - GOST
    - SAFER+,
    - Blowfish
    - ...

Demo

blowfish.c