computing: nature, power and limits—robotics applications (cis1.0)
fall 2006—lecture # G1
thursday 30-nov-2006

today:

- *catching up:*
  - the halting problem
  - computability and solvability
  - feasibility
- *new:*
  - security
  - encryption

the halting problem, computability, solvability, and feasibility

the halting problem

- a *loop* is a set of instructions that repeats several times
- we talked about 3 kinds of loops in RoboLab (counter-controlled, condition-controlled, forever) These concepts are the same in any computer programming language!
- here is an example in computer *pseudo-code*:

```
x=0;
do 3 times {
  add 1 to x
}
```

How many times does this loop execute?
What is the value of x when this code completes?

- another example:

```
x=3;
while ( x > 0 ) {
  subtract 1 from x;
}
```

How many times does this loop execute?
What is the value of x when this completes?

- and another example:

```
x=1;
while ( x < 5 ) {
  y = x;
}
```

How many times does this loop execute?
What is the value of x when this completes?

- a program containing an *infinite loop* will run *forever*—it will never HALT or TERMINATE
- this is called the HALTING PROBLEM in computer science—being able to look at a computer program and determine if it will ever halt (stop).
- sometimes, whether a program stops or not depends on *input* that it receives (like your robot receiving sensor input)
- here is an example:

```
myProgram( input: x ) {
  while ( x > 0 ) {
    add 1 to x;
  }
}
```

How many times does this loop execute if x = 0?
How many times does this loop execute if x = 1?

computability

- a problem is *computable* if it is possible to write a computer program that can solve it.
- a *non-computable* problem is also called *non-solvable*.
- the *halting problem* is not computable!
  i.e., can we write a computer program that will determine if any computer program and its input will halt?
  how would you answer this question?
  - could you try running the program? (what if it never halted?)
  - suppose we wrote a program ("A") that would take two inputs:
    another program ("P") and the input ("X") for the other program
    "A" works like this:
    if "P,X" halts, then "A" should run forever
    if "P,X" does not halt, then "A" should halt
  - the *paradox* is: what if we call program "A" on itself?
    the program cannot produce an answer!

- this is called *proof by contradiction*—we assume that a program does exist that can solve the halting problem; then we show that it cannot possibly exist.
- computability in general is an important question
- it was considered by concerned mathematicians even before digital computers were developed!
- in the 1930's, much work was devoted to this.
- the Church-Turing thesis (1940's) states basically that any computation that can be defined in an algorithm can be processed on a computer
- named after Alonzo Church and Alan Turing (really famous guy)

feasibility

- even if a problem is *computable*, it is not always *feasible* to write a program to compute it because sometimes it takes too long to solve a problem
- we talked about this with robot soccer:
  a robot might be able to find a ball using a complicated algorithm, but if it takes longer than 20 minutes to find the ball, the soccer game will be over! (extreme example...)
- could you write a program that could count the number of atoms in the universe (estimated at about 10^{80})?
  suppose it took 1 second to count one atom.
  how many seconds would the program need to run to count all of them?
  here's another way to look at it:
  how many atoms could the program count in a year?
  $num\_atoms\_per\_year = 60sec/min \times 60min/hour \times 24hours/day \times 365days/year$
  $= 31,536,000sec$
  $= 3x10^7$
  how does that compare to 10^{80}??

security, attacks, and cryptography,

why is there lack of security

- Internet and WWW computing standards (IP, HTTP, etc.) are public. *Security problems arise from the Internet and WWW being open and distributed systems.*
- The Internet is open and pervasive. Anyone may connect to it, and connections are everywhere.
- The Internet has many interconnecting components. e.g., a message from one computer to another will pass over many others, thus giving these others opportunities for espionage, etc.
- Web servers are extensible: they can be connected to all types of technologies (e.g., database servers) which may transform their functionality.
- Development speeds are very fast. Until recently, security has often been ignored in software development!
- HTML/HTTP were not designed for e-commerce but for document storage and transfer. e.g., browsers had limited functionality initially.

forms of attack

- Threats to integrity —
  e.g., an intruder modifies data stored in a database (like a credit card balance) or data is modified in transit.
- Confidentiality threats —
  Reading private data of a company or an individual (e.g., stealing credit card details)
- Denial-of-service threats (threats to availability) —
  Doing something which prevents a web-server from fulfilling its function; e.g., flooding a web-server with requests and thus overloading it.
- Authentication threats —
  Pretending to be someone who you are not

types of attacks

- Passive attacks
  - Eavesdropping on messages
  - Traffic analysis — An eavesdropper may be able to guess the content of a message by looking at the pattern of traffic; e.g., cold war warning systems
- Active attacks
  - Masquerades — Pretending to be someone whom you are not
  - Replay attacks — Capturing a data unit and retransmission for an unauthorized effect.
  - Modification-of-message attacks
  - Denial-of-service attacks

examples of attacks

- Low-tech attacks
  - Guessing passwords
  - Stealing passwords
  - Taking advantage of poor physical or clerical controls, e.g., bank employees creating fictitious accounts
  - Soliciting information from users which appears to be genuine but is not, e.g., pop-up windows masquerading as the on-line seller but are not
- Destructive devices
  - E-mail bomb — An email with a very large attachment, thus overwhelming the connection.
  - Denial-of-service attacks
    * Programs which spawn others, which in turn spawn others, etc.
    * The "ping of death": packets of data larger than permitted by TCP-IP often cause the receiving computer to crash.

viruses

- Malignant code which attaches itself to files resident on a computer. When run, it may delete or overwrite files, or send malicious emails.
- Types of viruses:
  - Executable virus: attaches to an executable file.
  - Data virus: infects a file containing data (e.g., a startup file)
  - Device driver virus: infects the device drivers of a system
- Anti-virus software — Usually looks for known viruses or for sudden changes in the size and content of files.
- Clever viruses work against anti-virus software — e.g., by mutating: changing their form or location ("polymorphic viruses") or by hiding themselves in other files ("stealth viruses")

other attacks

- Scanners
  - A program which detects security weaknesses
  - Usually developed to help human system administrators to find security flaws.
- Password crackers
  - Software which attempts to find passwords (e.g., by randomly generating all, or most likely, possibilities); often uses common words
- Sniffers
  - Devices which read the packets of data travelling on a network.
  - Usually used by system developers and administrators to assess system effectiveness.
  - May be able to detect passwords.
- Spoofing
  - Using one computer to masquerade as another, where the latter has privileged access to some third system

security requirements

- *Confidentiality:* information should not be accessible by unauthorized parties; protect against passive attacks.
- *Authentication:* make sure that the originator of a message or transaction is who they say they are; protect against masquerades and replay.
- *Integrity:* only authorized parties are able to change data, and only under pre-defined circumstances; protect against modification-of-message attacks.
- *Non-repudiation:* neither the sender nor the receiver of a message can deny that it took place.
- *Access control:* only authorized users are able to access resources within a system.
- *Availability:* The resources of a system are available to authorized users when they need to use them; protection against denial-of-service attacks

designing a secure system

- Assume two computers which wish to communicate over the Internet.
- Techniques for security have two basic components:
  1. Transformation of the information to be sent, e.g., putting a message into code before transmission.
  2. Some secret information shared by the two communicating computers but ot by any opponent, e.g., the code to be used for transformation.
- A trusted third party may be necessary to ensure secure transmission.
- Design a security service
  – Design an algorithm for transforming the information to be communicated.
  – Generate the secret information to be used with the algorithm. e.g., the coding scheme.
  – Develop methods for the distribution and sharing of the secret information, e.g., via a trusted third party
  – Specify a protocol to be used by the computers that uses the security algorithm and the secret information.

basic principles of cryptography

- Messages are put into code (encrypted) by the sender and decoded (decrypted) by the receiver.
- Basic ingredients of conventional cryptography:
  – Plain text input
  – Encryption algorithm
  – Secret key shared by sender and recipient
  – Cipher text (coded input text)
  – Decryption algorithm

cryptography example

- Suppose input text is
  THE SKY IS BLUE
- Algorithm:
  Replace each letter by the letter in the alphabet 1 step along.
- Key: 1
- Output:
  UIF TLZ JT CMVF
- A key of 2 would produce:
  VJG UMA KU DNWG
- The process used here is called *substitution* — substituting one element (in this case a letter) by another.
- Another process is *transposition* — Moving parts of the message around, e.g.
  TLZ UIF JT CMVF

requirements

- An encryption algorithm is needed. The sender must have this. The receiver must have a decryption algorithm, which undoes the encryption algorithm.
- Ideally, we would like a strong encryption algorithm, secure against attack. This is usually stated as follows: An opponent should be unable to decrypt the ciphertext or discover the key even if s/he is in possession of a number of ciphertexts together with the plain text which produced them.
- Both sender and receiver must have the secret key(s) for the process to work. This is the key weakness of symmetric encryption methods.
- Note: the security of conventional encryption depends on the secrecy of the key, not secrecy of the algorithm. It is usually assumed that it is impractical to decrypt a message on the basis of the ciphertext plus knowledge of the algorithm. We only need to keep the key secret.

classification of cryptographic systems

- The type of operations used to transform plaintext to ciphertext:
  - *substitution*
  - *transposition*
  - Usually some complex combination of these is used.
  - In any case, no information can be lost in the process.
- Whether sender and receiver use the same keys
  - *symmetric*: sender and receiver use the same keys
  - *asymmetric*: sender and receiver use different keys
- The number of keys used
- How the plaintext is processed.
  - A *block cipher* processes the input one block of elements at a time, producing an output block for each input block.
  - A *stream cipher* processes the input elements continuously, producing one element at a time as it goes along.

cryptanalysis

- the process of attempting to discover the plaintext or the key.
- some types of attacks:
  - known plain text attack:
    * The opponent has a sample of plaintext and ciphertext, and from this infers the keys; e.g., he may use brute force to try lots of different keys until successful.
    * Note that plain text may be compressed and may be numerical in origin, so brute force methods usually require some knowledge of the type of plain text used.
    * For a key of length 128 bits, it would take an opponent about 10^{18} years to crack!
  - chosen plain text attack:
    * The opponent gets the computer doing the encryption to encrypt some specially-chosen text (chosen to give clues about the ciphertext); e.g., the blocks may have many blanks or repeated words.
  - differential cryptanalysis attack:
    * The opponent gets the computer doing the encryption to encrypt several blocks of text which differ only slightly; e.g., the opponent then looks at the differences in the ciphertext.

some symmetric algorithms

- Data Encryption Standard (DES)
  - US Government standard in use 1977–1998.
  - Algorithm works on blocks of data (each 64-bits), and uses a 56-bit key.
  - Decertified in 1998 and replaced by 3DEA.
  - Possible to crack with brute force to find the key (i.e. key is too short).
- Triple Data Encryption Algorithm (3DEA)
  - Proposed in 1979, became a standard in 1999.
  - Applies the DES algorithm 3 times to plain text:
    Encrypt with Key A, decrypt with Key B, encrypt with Key C.
    Decryption is the reverse with the keys reversed: Decrypt with Key C, encrypt with Key B, decrypt with Key A.
  - Each key of length 56 bits, so effective key length of 168 bits.

symmetric key distribution

- the most important weakness of symmetric encryption.
- How do computers A and B agree their keys?
- Options:
  1. A selects key and delivers it physically to B.
  2. A third party C selects the key and delivers it physically to both A and B.
  3. If A and B have previously used a key, they could send a new key using an encrypted message.
  4. If A and B each have an encrypted connection to a third-party C, C could deliver a new key to each of A and B on encrypted links.
- For link encryption, options 1 and 2 may be feasible. But these are not usually feasible for end-to-end encryption.
- Option 3 is vulnerable: If an opponent ever finds a key, all future communications can be read.

asymmetric (or public) key algorithms

- Similar to symmetric key encryption, but we use at least 2 keys: One for encryption (the public key), and one for decryption (the private key).
- The steps involved are:
  - Keys are generated in pairs, a public key and a private key, by each person or computer (say Bob).
  - The public key is made public (e.g., on a web-site) by Bob.
  - Anyone who wants to send a message to Bob, uses his public key.
  - Bob decodes the message using his private key.
- This approach dates from 1976.
- Public key methods are not necessarily more secure than symmetric algorithms. But there is a larger computational overhead, and this makes brute-force attacks harder to execute successfully.

requirements for public key algorithms

- It is computationally easy for party B to generate a pair of keys.
- It is computationally easy for sender A to generate the cipher text on the basis of the plain text and the public key.
- It is computationally easy for party B to decrypt the resulting ciphertext using his private key and so generate the plain text.
- It is computationally infeasible for an opponent to determine the private key from the public key.
- It is computationally infeasible for an opponent to recover the original plain text from the public key and the ciphertext.
- In addition, we may require (not necessary but nice to have):
  Either of the two related keys may be used for encryption with the other used for decryption.

applications of public key methods

- Encryption —
  sending coded messages.
- Authentication —
  when we want to be certain that the sender of a message is actually the person (or computer) they say they are. The sender of the message uses his private key to encrypt the message. Only his public key will be able to decode the message.
- Digital Signature —
  The sender "signs" a message using his private key. This application is similar to authentication.
- Key Exchange —
  Two parties co-operate to exchange a session key, using the private key of one or both parties.

public key algorithms

- RSA algorithm
  - Developed by Rivest, Shamir & Adleman at MIT in 1977
  - RSA is a block cipher in which the plaintext and cipher text are integers between 0 and (n-1) for some n. If M is the plaintext number, and C is the cipher text number, the algorithm works as follows:
    - Encryption algorithm: $C = M^e \ modulo \ n$
    - Decryption algorithm: $M = C^d \ modulo \ n$
  - Both sender and receiver must know the values of n and e. The public key is a pair of numbers (e, n).
  - Only the receiver knows the value of d. The private key is the pair of numbers (d, n).
- Digital Signature Standard (DSS)
  - A standard agreed in 1993 for digital signatures in the American National Institute of Standards.
  - Only used for digital signatures (not for encryption or key exchange).

authentication using asymmetric keys

- Here we use the public and private keys in reverse order to encryption.
- Alice sends a message to Bob
  - Alice encrypts the message using her private key, which only she knows.
  - Bob receives the cipher text, and decrypts it using Alice's public key.
  - If Alice really did send the message, the output should be plain text.
  - If someone else (say, Mary) sent the message, then Alice's public key will not work on this message.
- So, this provides a way to authenticate a message, assuming the private key has not been stolen or somehow made public.
- Note that this method does not keep the message secret. Anyone can use Alice's public key (since it is public!) to decode Alice's message.
- This approach is also used for Digital Signatures, analogous to personal signatures on checks.

how to distribute public keys?

- Answer is simple: put on your web-site, email your friends, shout it from the roof-tops!
- But if Alice gets an email from Bob telling her that 1023 is his public key, how does she know it really is his? Maybe someone is impersonating him and sending out a false key in his name!
- Digital Certificates seek to get around this. A user (e.g., Bob) presents his public key to a trusted third party and receives a digital certificate. The certificate contains a public key together with a a user ID for the key owner (Bob), all signed by the third party.
- Examples of third parties: Government agencies or a bank. The user (Bob) can then give the digital certificate to anyone else (e.g., Alice).
- A standard for digital certificates is X.509.

public key distribution of symmetric keys

- How do 2 parties share a symmetric key? These are also called secret keys, to distinguish them from public and private keys.
- They could deliver them physically (e.g., by courier).
- If they already share a secret key, they could send the new one by encrypted message.
- They could use public key certificates, as follows:
  1. Bob sends Alice his public key using a public key certificate.
  2. Alice prepares a message.
  3. Alice encrypts the message using one-off symmetric key for this session, e.g., she uses a session key.
  4. Alice encrypts the session key using Bob's public key.
  5. Alice attaches the encrypted session key to the message and sends it to Bob.

  Only Bob is able to decrypt the session key (since only he has his private key). So, only Bob can read the original message.

location of encryption devices

- Where should the encryption occur:
  - End-to-end encryption — The entire message is encrypted
  - Link encryption — Each link along the way is encrypted; not as secure as end-to-end.
- Note, that end-to-end encryption can only encrypt the data portion (contents) of packets and not their headers, since the headers contain destination and sourcing information. Thus, the traffic pattern is not secure.
- Both methods can be used together. One key and method for the end-to-end encryption of data portion of packets. One key and method for the link encryption of the packet header.

cryptographic systems

Cryptography is used in several systems serving a variety of purposes

- Message Digest Functions
  - These produce a summary digest of a file, and can be used to see if the file has been altered.
  - Useful for detecting presence of viruses or tampering by opponents.
  - Sometimes used for message authentication codes (appended to a message, so that the receiver can see if the message was altered during transit).
  - Examples: HMAC, MD series (128 bit digest), SHA series (160 bit digest).
- Digital signatures
  - Unique identifier of a sender of a message
  - Can use public key cryptography in reverse
- Digital Certificates
  - Issued by trusted third party (e.g. bank, government agency) to verify user is who they say they are.
  - Usually third party's signature encrypted by the private key of the authorization party

    * So the receiver (Bob) needs to decode the authorization signature using the public key of the third party.
    * Then, if this works, use the public key of the sender (Alice) to decode her signature.

- PGP
  - "Pretty Good Privacy"
  - A publicly-available system for encrypting files and email messages
  - PGP uses:
    * RSA for management of keys in symmetric encryption
    * IDEA algorithm for sending data using symmetric encryption
    * MD5 scheme for ensuring no tampering.
  - Main weakness: if a public key is compromised, than a revocation certificate has to be issued to everyone in contact with the person whose keys are compromised.
- PCT — Microsoft product similar to SSL (Secure Sockets Layer, to be discussed later)
- SHTTP — A secure version of HTTP. Did not take off.
- SET (Secure Electronic Transactions) — for credit card info (more later)

some approaches to security

- Logging tools
  - software tools which monitor use of a computer which record particular events, e.g., user logins, transferring web-pages, attempts to access secure files
  - check for unusual events, e.g., access at unusual times, a user logging in and out repeatedly (could be seeking to gather info), a user mistyping a password (could be an attempted hack), a user accessing strange web-sites (e.g. military sites)
- Virus scanners — software which looks for unusual changes to files (especially operating system files)
- Security Checking software — used by system administrators to identify potential problems, e.g., scanners, password cracking software
- Network topology techniques
  - Design the topology of the network so as to make intrusion difficult
  - A common technique is a *firewall*
  - This is an extra element placed between a network (or a network element) and the external world.

firewalls

- Functions of a firewall
  - To monitor traffic into (and sometimes out of) a private network
  - To reject traffic which is considered suspicious or unauthorized
- Components of a firewall
  - Router (to monitor traffic into the private network, reject access from unauthorized users, reject or reroute rejected packets)
  - Bastion host, or proxy server (to provide a temporary store (cache) of pages held on a real web-server)
- Operation example
  - When a request for a web-page is received by the router from some client, it is accepted or rejected. If accepted, it is passed to the proxy server. If the page is in the cache, it is sent to the client who requested it. If the page is not in the cache, the proxy server requests a copy from the real web-server. When this is received by the proxy host, it is then sent to the client.

firewalls: what protection do they offer?

- Any attacker can only reach the proxy server — e.g., deleting web-pages from the proxy just deletes them from the cache, not from the real web server.
- A stronger form of protection is a screened subnet. This has 2 layers of protection, with routers on each side.
- In the central zone, between the two routers, are proxy servers for various functions:
  - A proxy web-server
  - A proxy email server
  - etc.

  Each contacts a real server in the internal network

firewalls: types

- *Packet-filtering router:* applies rules to each incoming packet and forwards or rejects them, one by one; filters may be on source or destination address fields.
- *Application-level gateway:* clients attempt to use a specific TCP/IP application, such as Telnet or FTP; To gain access past the firewall, they must enter access details; if details are correct, the application is allowed to proceed; otherwise, not.
- *Circuit-level gateway:* does not permit an end-to-end TCP connection
  - The firewall sets up two TCP connections: one between it and the external client; and one between it and the internal client. Messages are relayed across from one connection to the other. Normally, message are not examined – the security occurs in setting up the connections.
- Typically, a bastion host is the platform for an application-level or a circuit-level gateway.

firewalls: limitations

Firewalls cannot protect against everything!!

- e.g., disgruntled employees on the inside network
- e.g., attacks which bypass the firewall
  - e.g., internal systems may have a dial-out facility to connect to an ISP.
  - e.g., a modem to allow travelling employees to dial-in to the internal network remotely.
- Firewalls cannot screen every type of email or request
  - Viruses and malicious code may still get past a firewall, particularly if they are not presented in the usual form (e.g. email attachments).

Secure Sockets Layer (SSL)

- SSL was developed by Netscape for Netscape Navigator (its browser)
- It operates at the levels between: (1) HTTP and FTP and (2) TCP/IP
- Main functions of SSL:
  - SSL server authentication
    - ∗ Enables a client to confirm the identity of a server.
    - ∗ Uses public key cryptography to validate the digital certificate of a server and confirm that it has been issued by a valid certification authority.
  - SSL client authentication
    - ∗ Enables a server to confirm the identity of a client
  - SSL encryption
    - ∗ Uses symmetric encryption to send data to/from servers/clients.
- There are 2 sub-protocols
  - SSL Record Protocol — used for transmission of bulk data
  - SSL Handshake Protocol — used to establish the keys and algorithms to be used for data transfer

The SSL process

- Phase 1: Handshake
  - To authenticate server
  - To authenticate client (optional)
  - To agree secret keys and algorithms for part 2.
- Phase 2: Data transfer.
- SSL uses public key cryptography for the handshake, i.e.,
  - To authenticate client and server
  - To establish keys and algorithms for encryption of data transfer.
- SSL uses symmetric key cryptography for encryption and decryption of data in the data transfer.