Computer Architecture. Secure communication and encryption.

Eugeniy E. Mikhailov

The College of William & Mary

Lecture 28

Computer architecture

Type of hardware:

- biological, i.e. brain
- analog: electronic circuits, water flow, or even analog of sliding ruler
- digital computers

Operations system does not define computer architecture
MSDOS, Windows, Linux, BSD, MacOS, iOS, Solaris, Android, UNIX: who cares?
Though, historically several OSes are not hardware architecture agnostics for example MSDOS runs on x86 CPU only.

Loosely speaking type of CPU defines the architecture

Most common CPU lines

- complex instruction set computer (SISC)
  - x86 family (able to execute similar machine code at least for basic enough instructions, they do deviate at higher level commands. Search for extensions: MMX, SSE, etc):
    - Intel (8086, 80286, 80486, 80486, 80586 (Pentium) . . .
    - AMD some clones of early Intels (386,486) but then start to deviate, K6, Athlon, etc.
    - VIA
    - Cyrex

    First two manufactures are very common on a consumer market
  - Z80, PDP, VAX
- Reduced instruction set computer (RISC)
  - Sparc
  - DEC Alpha,
  - MIPS
  - Atmel AVR
  - ARM (think about your typical cell phone, Android, iPAD, iPOD, ebook reader)

Memory word

Typical number size for an integer number in arithmetic operation and addressable memory

- 8 bit (can **easily** address $2^8 = 256$ bytes)
- 16 bit (65536 bytes, i.e. 65kB)
- 32 bit (around 4 GB)
- 64 bit (1.8×10^{19})

Notice that more is not always better due to overhead. If all you need to add 2 digit numbers you do not want to spend time filling the rest of place holders with zeros.
Additionally not every OS can support large word size and will run your hardware in truncated regime. (You need to by special Windows OS to benefit from 64bit or recompile you Linux kernel and programs).

Thing to look when shop for new CPU

Before shopping decide what kind of typical problem you will solve: CPU intensive, memory intensive, can it be run in parallel, will you need to operate on batteries.

- Word size (see previous slide)
- CPU speed - how many FLOPS (floating-point operation per second).
  - if you do accounting you might not need floating points operation at all
- CPU cache size (memory on board of CPU), this days there are several layers of cache
- number of cores (CPUs) on a chip
- power consumption i.e. how hot and how long does it run

Byte order

Recall that now typical hardware unit is a byte which are assembled in machine words.
But what about ordering (endianness) of this word.

- Big-endian: similar to human notation the most significant digits go first (to the left): 1423. Example more resent ARM processors
- Little-endian: list significant bytes go first. Well known example of this is x86 architecture.
- There is also middle-endian: when order swapped between 32 and 64 bit representation, or other less bits word to more bits word.

When you exchange binary data between different architecture watch out for endianness.
Luckily `Matlab` stores its binaries in one particular endianness independent on architecture but it my result slightly longer read/write times.

Specially design CPU

Mars Science Laboratory - Curiosity rover

- CPU - RAD750 (radiation hardened)
- speed - 400 MIPS (Million instructions per second)
- memory:
  - 256 kB of EEPROM
  - 256 MB of DRAM
  - 2 GB of flash memory
- Power source - 100 W (after around 14 years).

Secure communication

You can try to exchange messages in secret, i.e. meet somewhere and make sure that no one listen to you. This is very realistic.
So we will focus on encryption and decryption: making our message unreadable for strangers (encryption) and the way to convert it back to a readable form (decryption).

Ways to encrypt

Very old method (traced back to ancient Greeks):

substitution cypher

For every letter use another letter.
For example: with a $\rightarrow$ z, b $\rightarrow$ w, l $\rightarrow$ e, m $\rightarrow$ a, and t $\rightarrow$ d

matlab $\rightarrow$ azdezw

Ways to encrypt

Very old method (traced back to ancient Greeks):

substitution cypher

For every letter use another letter.
For example: with a $\rightarrow$ z, b $\rightarrow$ w, l $\rightarrow$ e, m $\rightarrow$ a, and t $\rightarrow$ d

matlab $\rightarrow$ azdezw

Do not use in critical applications. To know why read about *frequency analysis* or read Arthur Conan Doyle “The Adventure Of The Dancing Men” (1903)

Things not to do

Never assume that your deciphering algorithm will be unknown to enemies!
If they find it you will have to change everything. Instead make it *secret key* dependent. If they now how your lock works they still need to find a lock.

Secret key algorithms

Symmetric key algorithm. In simple way: decide on a key (string on symbols) and add them receptively to your message.
Easy to break if you key is short but *impossible* if you key is longer the message and you never reuse this key. So called **one-time pad** method.
Hard to use in practice since we need to meet to exchange the pad.
Often very long keys are unpractical so a equivalent of a longer key is generated from a short one (this extreme oversimplification).

Public (shared) key cryptography

Based on numbers theory and the fact that some operations take very long time to do. For example, number decomposition to the primes.
You have *public key* and encrypt the message
I have *private key* and can decipher this message.
Example of use **https** protocol, also secure access to electronic mail servers (IMAP and POP) via TLS.
Additional benefits with some protocol you can sign your message so there is the proof that it was signed by you and that it is not changed (based on hash functions).

Some security software

Rule #1: Never, I repeat never use closed source security software

- PGP - Pretty Good Privacy
- GPG - GNU Privacy Guard

Above program allow you to encrypt, decrypt with shared key your mail or files, sign text messages, make a checksum of binaries.
Some software allows disk encryption but look for the Rule # 1