Trust in Computer Security

- Trust = Dependency or Reliance
  - If you must trust something/someone then misbehavior of that thing/person may impact on achieving your security goals.

- Example: Server running an webserver such as amazon.com

Transitive Trust

If A trusts B, and B trusts C, then A trusts C. It can be used to extend trust relationship.

Fighting against Transitive Trust:

- Limit number of different compilers (somehow thoroughly check them)
- Build it yourself
- Transparency
- Redundancy
- Debug the program interactively
- Limit developer's access
- Audit developers

Trusted Computing Base (TCB)

Sets of all hardware, firmware, and/or software components, you are trusting to achieve your security goal.

- Keep TCB as small as possible
- Automation
Fundamental goal of Computer Security

1. Confidentiality: concealment of information or resources
   - Access control mechanisms support confidentiality, e.g. cryptography.

2. Integrity: trustworthiness of data or resources
   - Usually phrased in terms of preventing improper or unauthorized change.
   - Includes *data integrity* (the content of the information) and *origin integrity* (the source of the data, often called *authentication*).

3. Availability: ability to use the information or resource desired

How can a computer support (mutually) distrusting users?

- Separate memory of different users

- Add “UserMode” register to CPU
  - Code must be running with UserMode = 0 in order to modify:
    - Interrupt Handler (IHA)
    - Clock & other interrupts
    - UserMode register
    - Page Table Register (PTR)
  - At boot, UserMode = 0
  - Before switching to execute an application, kernel changes UserMode ← 1
  - Interrupts set UserMode ← 0
- Kernel maintains page tables for each process
- Kernel sets PTR before switching to process
- Gives each process distinct physical pages
Communicate via shared physical pages

```c
runQuery(*key, userId){
    if (owner(*key) == userId)
        return value(*key);
}
```

- Problem: Time Of Check To Time Of Use (TOCTTOU/TOCTOU) error
  - A class of software bug caused by changes in a system between the checking of a condition (such as a security credential) and the use of the results of that check.