**Trusted Computing**
- “platform owner” is not trusted
  - Game: may modify software/hardware to cheat in online games
- Cloud computing apps distributed to many untrusted to many untrusted nodes
- Online movie/context distribution
- Closed box applications
  - Cell phone
  - Wireless driver
- Personal data management
  - Giving online app your info and how this info is used
- Online banking
  - High security app on low security architecture

**Security Goal of:**
- Manufacturer
- Other players
- Third party

**Outside discussion**
- Wireless network card’s power and frequency controls by drivers

**Terra**
e.g. game application

```
    s
   / \
  /   \ 
 c1    c2
```

**Control:** verify software running on c1

1. c1 says it is running certain software
   a. c1 can lie
2. c1 send a signature of the msg
   a. Still lie
3. c1 hash the app
   a. Install 2 copies and send correct hash
Tamper-resistant Hardware

\[ h_1 = \text{hash(CD1)} \]
\[ s_1 = \text{sig}_{s1}(h_1) \]

\[ h_2 = \text{hash(CD2)} \]
\[ s_2 = \text{sig}_{s2}(h_2) \]

\[ \text{key exchange } p_{c1}, p_{c2}, s_{c1}, s_{c2} \]
\[ \text{Game msg} \]

**Attack:** auto AIM proxy

**Solution:** encrypt, sign all game message

\[ c_1 = \text{sig}_{ca}(P_1) \]

1. Compute hash
2. Run software, generate session key
   a. \( P_{\text{game1}}, S_{\text{game1}} \)
   b. Hardware sign key \( s_1 = \text{sig}_{s1}(h, P_{\text{game1}}) \)

**Trusted Hardware**

1. \( S_{os} = \text{Sig}_{SW}(h_{os}, p_{os}) \)
2. \( h_{app} = \text{computed by OS} \)
   \[ \text{sig} = \text{signed using secret key of OS} \]
   \[ = \text{sig}_{so}(h_{app}, p_{app}) \]