How practical are BRUTUS & RVChecker?
- does one find enough errors using the combination of BRUTUS and RVChecker to make it worth using?
- Is the overhead of translating the protocol into specifications for BRUTUS and RVChecker too great?
  o There is no way to prove that the specification of the protocol remains consistent for BRUTUS & RVChecker
- Which ordering of tools is more practical?
  o RVChecker->BRUTUS: assumption->attack
    § Finds one attack
  o BRUTUS->RVChecker: attack->assumption
    § Finds an entire class of attacks

Problems with BRUTUS:
- understanding counterexample requires some previous knowledge
  o may be very complex

Other Tools:
- What other utilities/techniques are out there?
  o Model checking – more general technique
    ▪ Problem: the adversary in the model
      • brute force simplest to model
      • more powerful->more adversaries->more states
  o NRL
  o Murphi – 2 forms of adversary brute force & limited
    ▪ Brute force slower but still found error
    ▪ Limited – needed knowledge about protocol but not about flaws
- None of these tools define the exact syntax of knowledge
  o RV does not use semantic concept of common knowledge
  o Does not show complete theory generation rules
- Need to use trial and error for both RVChecker & BRUTUS
  o There is no guarantee you will hit upon correct combination of events to find error

Disillusionment:
- We already know about the flaws in these protocols
- More impressive to find new flaws
  o Murphi generating SSL3 from SSL2

Combining the two techniques
- High level RVChecker
  o Good to narrow down problem
  o Difficult to understand how assumptions turn into real attacks
- Lower level BRUTUS
  o Closer to design