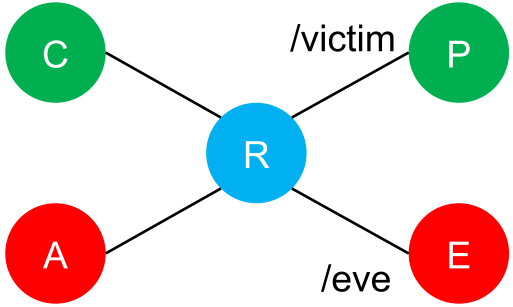
Mitigating Cache Poisoning from Link objects in NFD

- Link objects bring mobility support to NFD, at the cost of increased cache poisoning risk. Cache poisoning attack scenario:



Interest /victim/index.htm

Interest /victim/index.htm
Link delegate /victim to /eve

Data /victim/index.htm
bogus content

Mitigating Cache Poisoning from Link objects in NFD

- This project experiments with a mitigation solution:
  - Logically partition the PIT by Link objects: Interests with different Link are given distinct PIT entries.
  - In NDNLP header on Data, indicate the NDNLP sequence number of the corresponding Interest, so that downstream knows which Interests has triggered the Data reply.
  - Logically partition the CS by Link objects: Data with a certain Link object (or lack thereof) can only satisfy Interests with the same Link object, unless implicit digest is specified.
  - Extra benefit: use the Interest sequence number to speed-up PIT lookup

- You need:
  - C++11
  - knowledge about NFD forwarding pipelines and Link objects
  - Mininet or Mini-NDN or virtual machines on your computer to run 5x NFD instances

- Project demo: cache poisoning does not occur with the mitigation

- **Junxiao Shi**