Imperial College London

Developing teaching material for formal modeling of security protocols

Daming Chen and Michael Huth

Overview

- Undergraduate students learn concepts of and attacks on security protocols from coursework, but have difficulty applying these skills to designing and verifying real-world security protocols
- We currently develop teaching material to address this deficiency through hands-on modeling and verification of current security protocols using tools from academia/industry
- Material will be integrated into Networking course at Imperial College London, but also publicly available for adaptation or rearrangement

Approach

- Development of bespoke teaching material focusing on modeling security protocols
- Consists of individual flexible modules that can be rearranged to fit student and course needs
- Provides background material in related subjects that students may not be familiar with
- Includes supplemental material on more advanced topics for longer courses or experienced students
- Utilizes academic/industrial protocol modeling and verification tools

Goals

- Knowledge of at least one protocol modeling tool
- Competence in modeling security protocols
- Understanding of formal intruder models
- Ability to interpret tool results and attack traces
- Appreciation of theoretical limitations in formal modeling and verification of protocols
- Design and validation of novel security protocols

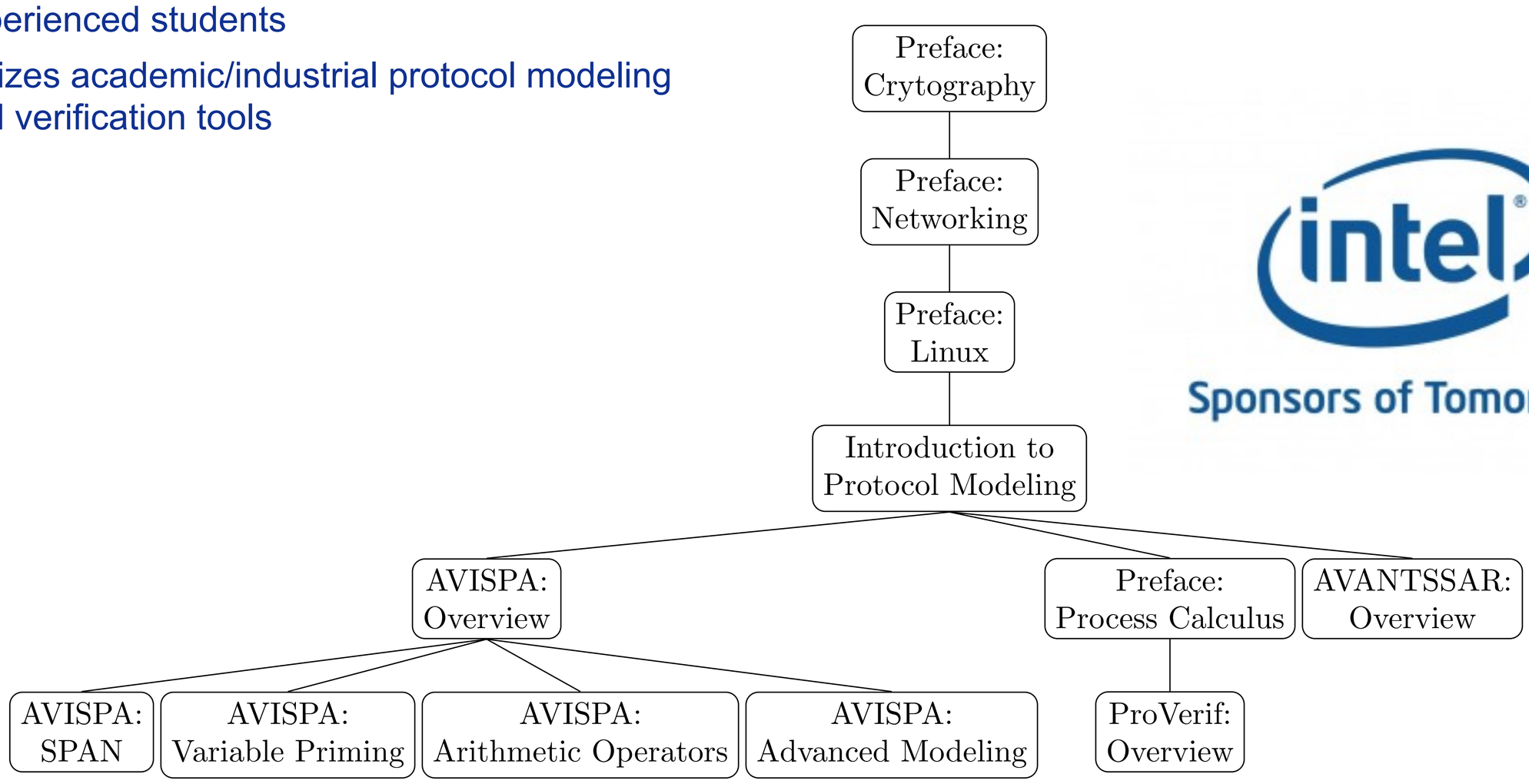

Fig. A: Proposed module sequence for teaching material

Example Protocol Model

```
role alice (A, B: agent,
    Ka, Kb: public_key,
    SND, RCV: channel (dy))
played_by A def=
local
    State : nat,
    Na, Nb: text
init
    State := 0
transition
    0.  State  = 0 /\ RCV(start) =|>
        State':= 2 /\ Na' := new() /\ SND({Na'.A}
_Kb)
                /\ secret(Na',na,{A,B})
                /\ witness(A,B,bob_alice_na,Na')
    2.  State  = 2 /\ RCV({Na.Nb'}_Ka) =|>
        State':= 4 /\ SND({Nb'}_Kb)
                /\ request(A,B,alice_bob_nb,Nb')
end role
...
role session(A, B: agent, Ka, Kb: public_key)
def=
    local SA, RA, SB, RB: channel (dy)
composition
    alice(A,B,Ka,Kb,SA,RA)
    /\ bob  (A,B,Ka,Kb,SB,RB)
end role
...
```

Conclusion

- Students need real-world understanding of modeling and verification techniques for security protocols
- Currently developing teaching material to build student experience through hands-on exercises
- Material will be integrated into Networking course at Imperial College London, but also publicly available
- Students will be able to interpret output of modeling tools and appreciate their theoretical approach