



FY2001 ONR CIP/SW URI



Software Quality and Infrastructure Protection for Diffuse Computing



Design and Analysis
of Protocols -
The Spycyce Way

Speaker: Joe Halpern
(Cornell)

SPYCE Objective: Scalable Distributed Assurance



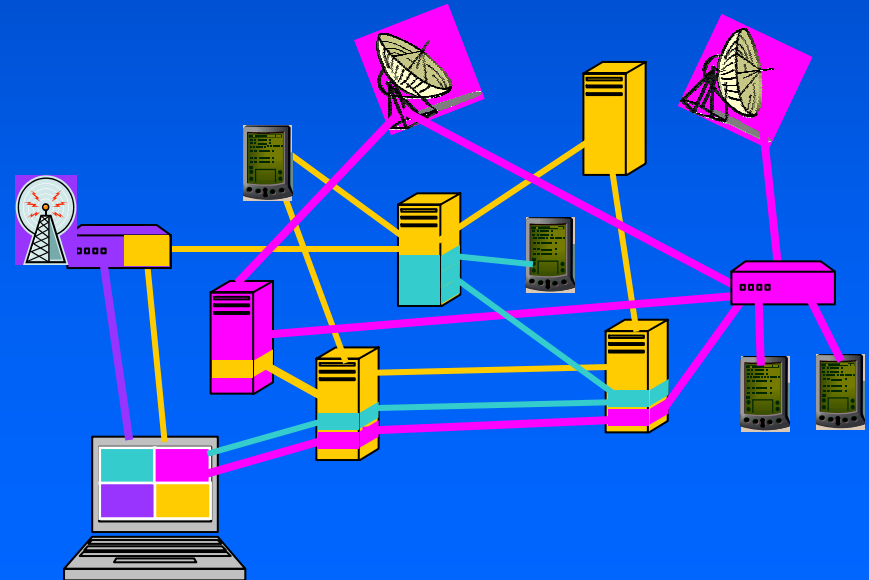
Develop fundamental understanding, models, algorithms, and network testbed, in order to reduce cost, improve performance, and provide higher reliability for networked operations across untrusted networks.

Incentives, Privacy, and Anonymity

Protocol Design and Analysis

Network Architecture

Trust Management



Distributed assurance requires

- good protocols
- good ways of verifying that they do what they're supposed to
- good ways of describing, analyzing, and specifying them

This can be hard!

Why Specifying Security Is Hard

Standard programming languages approach:

- Protocol = set of traces/executions/runs
- That's the SPYCE view too

But ... security protocols have special twists:

- Need to think about possible intruders, and what they can do
- What is a **fresh** message (nonce)?
- What is a **fair** protocol?
- What is **anonymity**?

Once we know how to describe/specify protocols, we can

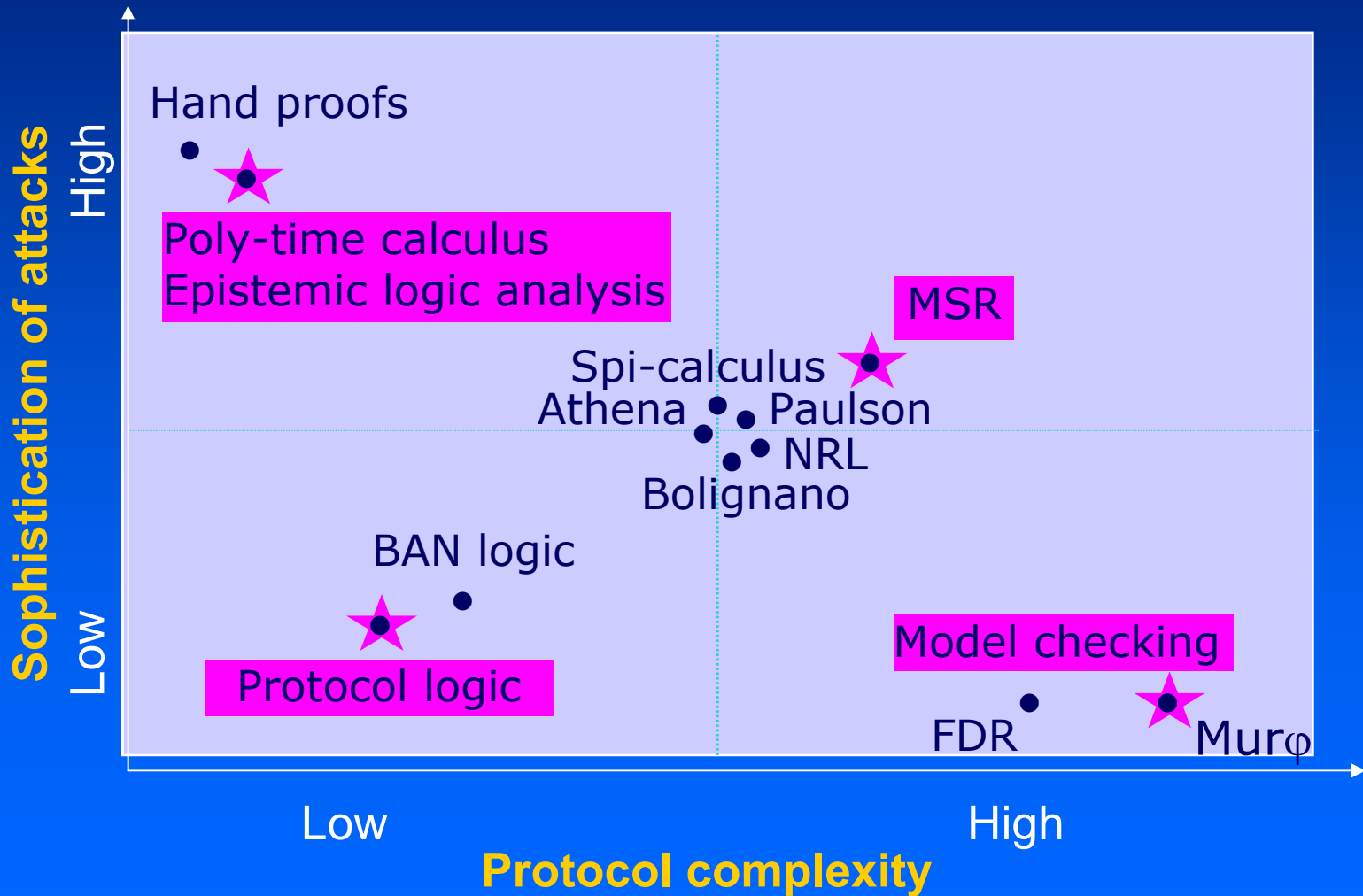
- prove specific protocols correct
- develop a deeper understanding of what makes things hard/easy
- build tools to help design/specify/verify protocols

We've made major progress in all these areas!

SPYCE Protocol Methods

- Model checking
 - Automated, finite state. Example: Mobile IPv6
- Multiset rewriting (MSR)
 - Symbolic computation, infinite state. Ex: Kerberos
- Epistemic logic
 - Understand who knows what; adversary capabilities
- Protocol composition logic
 - Axiomatic system; composition and design patterns
- Probabilistic poly-time process calculus
 - Full computational model; compositional reasoning using observational equivalence

Protocol analysis spectrum



*SPYCE Accomplishments:
A Representative Sample ...*

A. Using MSR: Kerberos

[Butler, Cervesato, Jaggard, Scedrov]

- Kerberos: real world protocol
 - Repeatedly authenticate a client to multiple servers
 - Minimize use of client's long term key(s)
- Results
 - Formalized Kerberos 5 at different levels of detail
 - Observed anomalous behavior: some properties of Kerberos 4 do not hold for Kerberos 5
 - Proved that various properties do hold
 - Shows power of MSR as specification language
- Interactions with Kerberos working group

Using MSR: Contract Signing

[Chadha, Mitchell, Scedrov, Shmatikov]

- Need third party to arbitrate contracts, in worst case
- Optimistic protocols
 - Can complete contract without third party
- **Impossibility Theorem:** There is no timely, fair, optimistic, and balanced contract signing protocol.
 - If protocol is timely, fair, and honest, dishonest player can always choose the outcome unilaterally
 - Bad news for e-commerce

B. Epistemic Logic

Knowledge plays a fundamental role in security:

- What does an intruder know?
 - Can she factor?
- Anonymity:
 - Can an observer know who performed an action?
- Noninterference:
 - What does unprivileged user know about privileged user's state?

Modeling an Intruder

[Halpern, Pucella]

- The literature focuses on “Dolev-Yao” intruders. But what about
 - an intruder that guesses?
 - an intruder with side information?
- Need to talk about what intruders can't know due to resource limitations
- Can use **algorithmic knowledge** (algorithms that model intruder's reasoning) to do that
 - Different algorithms -> different intruders

Capturing Security Concepts Using Knowledge

[Halpern, van der Meyden, Pucella]

- What's a "fresh" message?
 - One that's unpredictable
 - No one **knew** its content in advance
- What's a "good key"?
 - One that only authorized users **know**.

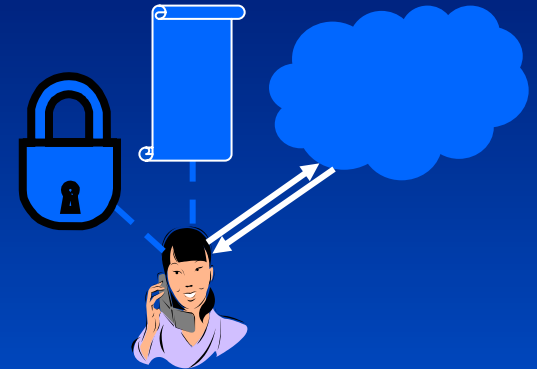
We need to make sense of these notions to prove that protocols are correct

- Can do this formally using knowledge

C. Protocol Derivation Logic

[Datta, Derek, Durgin, Mitchell, Pavlovic]

- Reason about agent's **knowledge**
- Protocols are constructed from components by applying:
composition, refinement, transformation
- Protocol logic supports derivation
 - General composition proofs
- Applied to several protocol families
 - STS, ISO-9798-3, JFKi, JFKr, IKE



D. Probabilistic Process Calculus

[Lincoln, Mitchell, Scedrov, Ramanathan, Teague]

- Probabilistic polynomial-time execution model
- Specify security via equivalence to “ideal” protocol
- Also state cryptographic assumptions via equivalences
- Leads to new proof system
 - Equational reasoning
 - Based on probabilistic bisimulation, asymptotic equivalence
- Applications
 - Characterize computational indistinguishability
 - Proof of semantic security from computational assumption (both stated as equations)

Summary

- We use a number of approaches for representing protocols, but they all boil down to sets of runs
 - We speak the same language
 - Can transfer results between frameworks
- We've examined specific protocols
 - Kerberos, contract signing, BGP
- ... and proved general results about classes of protocols
 - impossibility results: no fair optimistic protocol

- Central role of knowledge
 - in specifying properties of interest of fairness, anonymity
 - for defining basic concepts of freshness, goodness of keys
 - in characterizing intruders

(Some) Next Steps

- Further investigation of practical protocols
- Synthesis of various approaches
 - move to upper right of protocol analysis spectrum
- Automating verification
- Adding utilities to specifications
- Verifying mechanisms
 - mechanism = set of rules for playing a game, designed to encourage “good” behavior
 - o e.g. tax system, type of auction