Works so far:

1. Geometric Logic
2. The Chase
3. Normalization, Efficiency, Chaining

Work to do:

1. CPSA algorithm
2. Translating protocol specs into geometric logic
3. Extensions

1 Motivation

What are the advantages of geometric logic; why bother performing cryptographic protocol analysis in this new domain?

Foremost, cryptographic protocol analysis is traditionally implemented in imperitive programming languages. Switching to a declarative style has the advantage of making correctness easier to verify, and the system easier to reason about in general. First order logic is also a natural way to express the definitions of the formalism used.

Second, a geometric theory is both flexible and extensible. Any extension expressible in geometric logic can be added trivially; the only implementation concern is the priority of the rules.

Third, the chase can perhaps shed some light on the working of existing algorithms. It seems to mimic them in a way; there is perhaps a close relation between the chase's search for minimal theories and the algorithms' search for representative shapes.

2 Geometric Logic

A run of the chase *fully terminates* when it returns a finite number of finite models and then halts.

3 Implementation

3.1 Structure

The geometric theory is partitioned into three sections:

1. Protocol Specification
2. Strand Space Axioms
3. Inference Rules

4 Target Theorems

1. Show that every model generated by the chase is, in fact, an infiltrated skeleton.
2. If A is a realized infiltrated skeleton, then there exists a homomorphism from some model returned by the chase to A.
3. Show that the chase fully terminates under certain conditions.

[Possible proof technique: assume the theory ends with "completeness rules" – the rest of the axioms of the strand space formalisms – and then show them to be unnecessary.]