

# What is CryptoForma?

Eerke Boiten

# CryptoForma: the objective

- Getting formal methods to work for modern cryptography: probability, approximate correctness, security properties, attack models
- “FM people” and “crypto people” and a few who really straddle; academia and industry

# CryptoForma: the details

- A proposed UK network of excellence, starting with Birmingham, Bristol, HP, Kent, Microsoft, Newcastle, Royal Holloway,... (not closed!)
- EPSRC: initially rejected for novel formalistic reasons
- Now being considered (finally/again)
- 3 years: 1 workshop/year, 3 meetings/year, visits, tutorials, ...

# This meeting ...

- Many thanks to Microsoft for hosting and sponsoring
- And to Andy, George and Rachael for organising
- “Now what” at the end? Volunteers for a next one?

# A Bottom-Up Formal Methods Perspective

Eerke Boiten

CryptoForma meeting #0 at  
Microsoft Cambridge, January 2009

# Overview

- 4 reasons for not bothering
- Definitions and dogmatism
- A bottom-up perspective: a few positive developments towards ...

# 1. It's hard!

- I don't understand many of the proofs of security in crypto papers.
- Dan Grundy's talk: probability theory and complexity theory have no strong tradition of symbolic reasoning, ...
- *Some* cryptologists also don't understand *some* of the proofs of security in crypto papers.
- Conference- biased research culture ...
- Smart people are easily bored.

## 2. It's supposed to be impossible

- “You can't do refinement for security”
- E.g.

... System output is underspecified in a particular error case?

Refine to ...

- ... System output gives away my pin code!
- Morgan “The shadow knows” (MPC 2006)

# 3. The culture clash in TCS

- There are two kinds of theoretical computer science: semantics and complexity
- Semantics is algebra and logic and clean compositional formalisms, proofs are symbolic and positive and machine-verifiable
- Complexity is reduction and Turing machines, proofs are by contradiction, witnesses are pulled out of hats
- I'm on one side, crypto is on the other ...

## 4. It's all been done anyway

- Paulson, and Schneider/P.Ryan/Lowe/... have shown that for perfect security in an abstract setting, non-determinism subsumes unbounded malevolence.
- Computationally sound logics and computational Dolev-Yao ...

# Some definitions

- *Formal specification*: describing all required properties of a product (in advance)
- *Verification*: checking (formal, automated?) that product satisfies specification (post-hoc?)
- *Derivation*: systematic construction of product from specification, ensuring verification trivial
- *Refinement*: preservation of all properties of interest (*as a relation, as a process*)

# A bit of Dutch dogmatism

In decreasing order of desirability:

- Derivation
- Post-hoc verification
- Verification checking
- Informal verification
- Post-hoc establishing specification

With machine support / automation

# Action refinement

- Protocol: single objective, multiple steps
- Can be done in process algebras but ...
- Action systems, TLA: stuttering steps
- ASM: big step diagrams
- Recent results on non-locking, interference, completeness of simulation (Hesselink, Schellhorn, Groves, Banach)

# Approximate Refinement

- Arose to refine  $\mathbb{Z}$  by `int`, etc (and as an alternative to “retrenchment”)
- Needed as perfect security is often unrealistic or even provably impossible (commitment)
- See [Boiten & Derrick, ZB 2005] or a draft paper on reconstructing commitment
- Mingsheng Ying in a probabilistic setting

# Probabilistic refinement

- Nonces, and attack models
- Specification formalisms difficult: how does probabilistic choice interact with other choice?
- McIver and Morgan for action systems (inspired Hehner, Schneider&Robinson, Kleene algebraists)
- Probabilistic CSP still not quite solved

# The final questions: Attack models

- Non-determinism subsumes malevolence, how does that generalise to include guessing?
- ... and how is that restricted to remain polynomial? (Is timing termination?)
- What does refinement calculus say about game-hopping?
- Can we encode things in relational ADT rather than action systems or CSP?