Interaction vs. Automation of Tools: Issue or Speculation?

A pretext to present the Inductive Method's recent advances

Giampaolo Bella

Università di Catania, Italy
De Montfort University, UK

FMATS Workshop — 7th December 2011
Microsoft Research Cambridge

Interaction vs. Automation?

Personal Experience

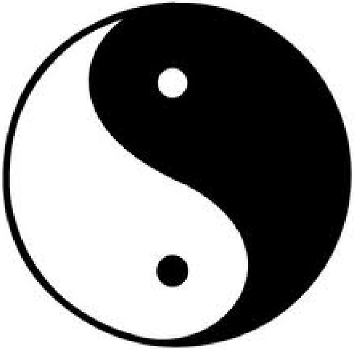
- AVISPA (SATMC)
  - + Rather fast; can switch MC; integration with testing
  - – Spec entangled without abstract language; explosion
  - – Took months to clever student; liaison with developer
- ProVerif
  - + Quick start
  - – Termination; timestamps; linkability?
- ATP (first-order)
  - + Quick start
  - – Obvious language limitation
- ITP (higher-order)
  - + Flexibility; proof reuse; good automation
  - – Slow start; lack of tutorials; "specialised skills"?

Personal Experience

Practitioners show thirst for theoretical foundations

- AVISPA: rule-based languages
- ProVerif: process algebras
- ATP: decision procedures
- ITP: conditional term-rewriting

Interaction vs. Automation



Claim

No tool seems to make any exception

How Much Interaction — officially

Inductive Method

- Tool Development
  - Rather slow
- Specification
  - Interaction: low
- Verification
  - Interaction: high

SATMC

- Tool Development
  - Continuous
- Specification
  - Interaction: low
- Verification
  - Interaction: low or 0

When facing case studies that are

- Traditional: proof reuse (IM), spec reuse (SATMC)
- Innovative: frequent spec update to face explosion/ non-termination (SATMC), new proof strategy (IM)

How Much Interaction — actually

Inductive Method

- Tool Development
  - Rather slow
- Specification
  - Interaction: low
- Verification
  - Interaction: X

SATMC

- Tool Development
  - Continuous: Z
- Specification
  - Interaction: Y
- Verification
  - Interaction: low or 0

Empirical Observation

X appears to be near $Y + Z$

Recent Applications of Inductive Method

- Physical properties (Basin et al.)
- Multicast protocols (Martina)
- More threat models (me)
- Protocol composition (Butin, Gray and me)
- Privacy (Butin and me)
- Security ceremonies (Coles-Kemp and me)

Claim

Fexibility to overrule debate on interaction vs. automation

Physical Properties

- Communication constraints, such as influence of communication medium and distance, on travel time
- Properties such as authenticated ranging (Charlie's distance in presence of Eve) and distance bounding (Eve's distance)

Example

```
lemma authranging_secure:
  assumes
  rang: tr ∈ ranging and aneqb: A≠B and
  believe: (t, Claim (Honest A)
                 ⦃Agent (Honest B), Real d⦄) ∈ set tr
  shows d ≥ pdist (Honest A) (Honest B)
```

Multicast Protocols

- General message-sending primitive to also account for unicast and broadcast as extremes
- Can tackle new class of protocols, such as for electronic auctions, and related properties

Example

```
theorem bid_secrecy:
⟦Multicast B mc_group (λC. ⦃Nonce aid,
 Crypt (pubK C) ⦃Nonce (share (nat t, mc_group, C)
                          ⦃Agent B, Nonce v, Nonce w⦄),
                   Nonce aid⦄,
 ...  ...
 B ∈ bad; Spy ∈ set mc_group; evs ∈ fr⟧
⟹ Nonce v ∈ analz (knows Spy evs)
```

More Threat Models

- General Attacker: anybody — more than one agent at a time and with personal interests — may deviate from protocol (was super-tough with SATMC)
- Supports: analysis of multy-party independent attacks (competition); evaluation (conflict); assumption elicitation

Example

lemma *secret_parts_agent:*

$$m \in parts\ (knows\ C\ evs) \implies m \in initState\ C \lor$$
$$(\exists A\ B\ X.\ Says\ A\ B\ X \in set\ evs \land m \in parts\{X\}) \lor$$
$$(\exists Y.\ Notes\ C\ Y \in set\ evs \land m \in parts\{Y\})$$

Protocol composition

- (Mutually-)dependent inductive definitions: simple (yet general?) account for protocol stacking, sequencing, or inteleaving
- Analysis of, e.g., multi-protocol attacks or properties

Example

```
| NS2:
 ⟦evs2 ∈ ns_public; Nonce NB ∉ used evs2;
  Says A' B (Crypt (pubEK B) ⦃Nonce NA, Agent A⦄)
     ∈ set evs2;
  evscb ∈ cert;
  Crypt (priSK CA) ⦃Key K, Agent A⦄
     ∈ parts(knows B evscb)⟧
⟹ Says B A (Crypt K ⦃Nonce NA, Nonce NB, ...  evs2
```

Privacy

- Unlinkability: operational inspection of traces
- *aanalz*: association analyser by observer; *asynth*: association synthesiser over linked associations

Example

```
theorem foo_V_privacy_aanalz:
⟦Says V Adm ⦃Agent V, Crypt (priSK V)
            (Crypt b (Crypt c (Nonce Nv)))⦄ ∈ set evs;
 a ∈ aanalz Spy evs;
 Agent V ∈ a; V ∉ bad; V ≠ Adm; evs ∈ foo⟧
⟹ Nonce Nv ∉ a
```

Example

```
inductive_set
 asynth :: msg set set ⇒ msg set set
 for as :: msg set set where
  asynth_Build [intro]:
  ⟦a1 ∈ as; a2 ∈ as; m ∈ a1; m ∈ a2⟧
   ⟹ a1 ∪ a2 ∈ asynth as

theorem foo_V_privacy_asynth:
⟦Says V Adm ⦃Agent V, Crypt (priSK V)
           (Crypt b (Crypt c (Nonce Nv)))⦄ ∈ set evs;
 a ∈ asynth(aanalz Spy evs);
 Agent V ∈ a; V ∉ bad; V ≠ Adm; V ≠ Col; evs ∈ foo⟧
⟹ Nonce Nv ∉ a
```

Security Ceremonies

- Security may fail in reality despite correct technology
- Ceremony as a protocol with outer layers: O.S., HCI, Personal, Communal
- Room for analysis at each layer — hierarchical?

Example

```
theorem U_registers_without_confidence:
⟦((U,Registers,P), sigma) ∈ set evs;
  ∀ sigma'. ((P,Explains,U), sigma') ∉ set evs;
  evs ∈ ceremony⟧
⟹ Confidence ∉ sigma
```

Conclusions

- All tools seem to exhibit some levels of interaction and automation, distributed through the phases of development, specification and verification
- A number of recent security applications, each potentially requiring novel specification and verification effort, currently need formal analysis
- Flexibility, as the simplicity in coping with new applications, seems to acquire relevance over debate on interaction vs. automation
- The current generation of the Inductive Method appears to be useful in this scenario