The Applied Pi Calculus... with Proofs

Bruno Blanchet

INRIA Paris-Rocquencourt
Bruno.Blanchet@inria.fr

joint work with Martín Abadi and Cédric Fournet

July 2015
The applied pi calculus

- Designed by Abadi and Fournet (Mobile Values, New Names, and Secure Communication, POPL’01).
- Extension of the pi calculus with terms instead of names for messages.
- Language for modeling security protocols:
  - Terms represent protocol messages.
  - Function symbols represent cryptographic primitives.
  - The properties of these primitives are modeled by equations.
  - The input language of ProVerif is a dialect of the applied pi calculus.
- The applied pi calculus and ProVerif are widely used.
  - Interesting to make them converge, with a solid theoretical foundation.
Our contribution

- Minor changes to the language
  - Closer to ProVerif
  - Channels are any term, not just names
  - A single sort Channel for all channels, instead of Channel\(\langle \tau \rangle\)
- Detailed proofs of all results

Theorem

1. Observational equivalence is labelled bisimilarity: \(\approx = \approx_I\).
2. Two labelled semantics: simple labels \(\nu x.\overline{N}\langle x\rangle\); refined labels \(\nu\tilde{x}.\overline{N}\langle M\rangle\).
   The labelled bisimilarity is unchanged: \(\approx_I = \approx_L\).

- Minor fixes; some side-conditions were not explicit
- Introduced a notion of partial normal form \(\nu\tilde{n}.(\{\tilde{M}/\tilde{x}\} | P)\) for processes, useful for many other proofs
- 72 pages of proofs...
Indifferentiability is a security notion for hash functions, in the computational model.

The following two systems are indistinguishable:

1. The real hash function, defined from a compression function considered as a random oracle.
2. An ideal hash function modeled as a random oracle; the compression function is defined from the hash function.

We show the corresponding property in the symbolic model:

- the two systems are observationally equivalent
- for a particular construction: chop Merkle-Damgård
  
  1. iterate the compression function;
  2. return the first $n$ bits of the result.
Conclusion

- Importance of **detailed proofs**.
- With the minor changes we made, one should be able to show that
  - The plain processes of the applied pi calculus are a subset of the **ProVerif** input language.
  - The semantics and the notions of observational equivalence match.
- If you would like to read a draft, please contact me.