ETAPS Daily

Issue 6

Thursday, March 21st, 2013

A new "Pope" for ETAPS

The banquet of yesterday night was the last official event where Prof. Vladimiro Sassone acted as ETAPS SC chair. After 6 years of intense and incredibly productive work, he left this important role to his colleague Prof. Joost-Pieter Katoen. Joost-Pieter has been one of the leaders of ETAPS for the past years, both in his official role as the publication chair and in several other tasks he took upon himself to promote and facilitate the running of the conferences. From the scientific aspect, he is a well known researcher, with wide international recognition. He has published in many of the ETAPS conferences and have been the program chair of several (also non-ETAPS) conferences. While thanking Vladimiro for all his efforts and the results he obtained, we all wish all the best to the new chair that, with his enthusiasm and boundless energy, will surely address many of the issues that ETAPS is facing now, as well as those that will come up in the future. We wish him to promote harmony among the bodies that ETAPS encompasses as well as to enable ETAPS to maintain its status as one of the top venues in the field.

A thematic journey into Rome: Churches

There are more than 900 churches in Rome, most of which are Roman Catholic. It is known that in 336, Pope Julius I had set the number of presbyter cardinals to 28, so that for each day of the week, a different presbyter cardinal would say mass in one of the 4 major basilicas of Rome: *St. Peter*, *St. Paul outside the walls*, *Santa Maria Maggiore* and *St. John Lateran*. These four basilicas had no cardinal, since they were under the Pope's direction. Traditionally, pilgrims were expected to visit all four basilicas; together with *San Lorenzo outside the walls*, *Santa Croce in Gerusalemme* and *San Sebastiano outside the walls*, they constituted the Seven Pilgrim Churches of Rome. In the 2000 Jubilee, the 7th church was instead *Santuario della Madonna del Divino Amore*, as appointed by Pope John Paul II.

The oldest churches date back to the 4th century; however, over the centuries, most of them have undergone various changes. Thus almost all of them appear more recent and usually merge together elements of different periods and styles. Among the others, we want to mention the beautiful churches of the Celium hill (including the round church of *Santo Stefano Rotondo*). Several churches are also famous for the masterpieces they contain: e.g., *San Pietro in Vincoli* hosts the Moses statue by Michelangelo, *San Luigi dei Francesi* and *Santa Maria del Popolo* a few beautiful paintings by Caravaggio, *Santa Maria sopra Minerva* with the Carafa Chapel by Filippino Lippi and the Christ statue by a young Michelangelo. But every church features some unique treasure it is worth discovering during a walk through Rome.

Today's program:

Timetable:

9^{00}-10^{00}: FASE invited talk
10^{00}-10^{30}: coffee break
10^{30}-12^{30}: parallel sessions
12^{30}-14^{00}: lunch
14^{00}-15^{00}: TACAS invited talk
15^{15}-16^{15}: parallel sessions
16^{15}-16^{30}: coffee break
16^{30}-18^{00}: parallel sessions

Scientific Events:

Invited talks (Czamecki, Grumberg): Aula Magna (campus)
CC: room A1
ESOP: room B2
FASE: room A2
TACAS: room B1
TACAS SW competition: room A1

Other Events:

FASE SC meeting: lunch time, room A2
SC meeting: Restaurant *Il Tunnel*

Weather forecast:

	9-13	13-17	17-22
Thu			
Fri			
Sat			

SAPIENZA UNIVERSITÀ DI ROMA

An interview with the two Unified Speakers

This year, the two unified speakers are from the field of security and have reported their different approaches to the verification of security protocols. In the morning, we had Gilles Barthe, speaking about Computer-aided cryptographic proofs; in the afternoon, we had Cedric Fournet, speaking about his implementation of TLS 1.2 with verified cryptographic security. We now report here a short discussion we had with the speakers after their talks.

In your invited talk you suggested to consider cryptography as part of the broader discipline of program verification rather than of computer system security. Would you care to elaborate on that statement?

What I was trying to say is that you can actually think about provable security as some non-standard form of program verification. Program verification is a good match to solve the problems they have in security. I think there are very tight connections between important concepts in programming languages and program verification and what cryptographers actually do. For example, there are relationships between the notion of simulation that they use and the notion of simulation that programming language people are working on.

How can programming language techniques help cryptography proofs?

I think there are many areas from programming language that could be applied in very beneficial way to cryptography. One is invariant generation, which is currently a major hurdle in computer aided cryptographic proof. There is some nice work on generalization of observational equivalence and decidability results for probabilistic observational equivalence that could be very useful. In our work sometimes you invent new technologies, but it's also very often a question of merging existing technologies such as we do with probabilistic process algebra and relational Hoare logic. Another thing we are doing is using these synthesis techniques which I mentioned in my talk and this domain specific logic to build a verified atlas of cryptographic schemes.

Which is, in your opinion, the main difficulty in verifying protocols? Their informal specification? The model of the adversary? The assumptions on the crypto-primitives?

Given a protocol description, a first difficulty is to arrive at a reasonable formal security specification: what are the goals of the protocols? what are the assumptions about the adversary for each of these goals? In informal specifications for communications protocol, this is often missing, or phrased in high-level terms---the main purpose of those specifications is to enable different implementation to interoperate, rather than facilitate their security analysis. Besides, protocols are often used for purposes different than those initially considered by their designers. For instance, the TLS standard says “the protocol allows client/server to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery”, but it does not specify how clients and servers actually control the protocol, and which security events to consider for authentication and secrecy. In many cases, security specifications emerge only as new classes of attacks are found.

How many attacks still not discovered do you think are contained in the protocols we use every day?

I have no idea! My guess is that most of the remaining attacks are around these protocols (in their implementation details, their configuration, or their long-term key management) rather than in their core cryptographic design.

Usually, verification methods relying on types, to statically ensure safety, are over-restrictive. Do you really think this is the right methodology to use, even with your probabilistic enhancements?

This is a fine method for the modular automated verification of ML implementations of protocols, but there are many alternatives. For example, ProVerif provides much better support to prove properties and find counter-examples, without the need for type annotations; interactive proofs with EasyCrypt offer more flexibility to deal with complex cryptographic schemes; and tools such as VCC or VeriFast are called for to verify low-level C implementations.