Our research area

The history • The key people, publications • Examples

Some requirements to be noted

* An overview of the history of the research area.
* The information about each of the sub-areas you have chosen to cover.
* Some of the key researchers, publications and results.
* While the website is not exclusively about the research done at UCL, highlight at least some of the current research being done here.
* Links and references to further information and cited work.

NOTICE: highlight the names of all the researchers (might be included in key people section) in red and highlight the titles of all the publications/workshops/papers… in blue.

History Overview

\*

Around the beginning of academic year 2012-2013: UCL PPLV was established. First seminar held on 31st Aug 2012 by Delphine Demange was about ‘A Buffered Memory Model for Java’.

Highlight Research Areas

1. **Mechanized program verification**

This is triggered by practical advances where in special domains --- in avionics code, in device drivers, and other small systems programs --- we have seen fully automated verification of selected properties such as memory safety and termination and API protocol conformance. E.g. Separation Logic and the Space Invader tool, and on SLAM and Terminator.

1. **Automatic verification**

Mathematical logic is moving from being a conceptual mathematical tool to a collection of techniques which can be used by practising engineers and scientists in solving the concrete problems that they face.

1. **Systems and security modelling, including work on access control policies and models**

Techniques from semantics (e.g., process algebra) and logic (e.g., modal and substructural) have been developed to provide a basis for systems modelling tools, such as Gnosis, that deploy these ideas in the style of classical mathematical modelling --- as practised in engineering, economics, and so on --- to understand the interaction between system architecture and security policy.

1. **Symbolic model checking**

Key People

**UCL PPLV researchers (4 in total, listing 2 that might be key ppl)**

**Prof Byron Cook (**[**https://iris.ucl.ac.uk/iris/browse/profile?upi=JBCOO12**](https://iris.ucl.ac.uk/iris/browse/profile?upi=JBCOO12)**)**

o Principal Researcher at Microsoft's laboratory at Cambridge University

### o Research interests include program analysis/verification, programming languages, theorem proving, logic, hardware design, operating systems, and biological systems. Recent work has been focused on the development of automatic tools for proving program termination/liveness, memory safety, as well as properties about models of biological systems.

**Prof Peter O'Hearn (**[**https://iris.ucl.ac.uk/iris/browse/profile?upi=PWOHE10**](https://iris.ucl.ac.uk/iris/browse/profile?upi=PWOHE10)**)**

o Part-time professor of Computer Science, engineering manager at Facebook, 2016 Gӧdel Prize winner for discovery of concurrent separation logic (annual prize for outstanding papers in theoretical computer science), fellow of Royal Academy of Engineering in 2016

o Separation logic – verifying memory-aware programs, invented early 2000s (pioneered by Peter O’Hearn and John Reynolds), led to a career working at Facebook running the Facebook Infer which analyses code to intercept critical bugs, help prevent crashes and poor performance.

Research Projects

1. **Boosting Automated Verification Using Cyclic Proof**

The recent hard-coded predicates result in reduced program coverage and increased rates of false negatives. Thus, methods for reasoning with \*general\* inductively defined predicates could greatly enhance the state of the art.

Cyclic proof, in essence, implements reasoning by infinite descent à la Fermat for general inductive definitions. This makes cyclic proof adding inductive reasoning capability, for general inductive predicates, to the many components of an interprocedural program analysis an attractive method for automatic proof search.

1. **Resource Reasoning**

Generally, for all kinds of computer system, it is second nature for a programmer to think carefully about the way the system handles resource as part of the programming task in hand. The point of view of this research is that there can be a novel kind of mathematical theory, which we call resource reasoning, which goes hand in hand with programmers' informal thinking about resources.

# **Compositional Security Analysis for Binaries**

This project started on 26 November 2013 and ended on 25 November 2016. The research team from University College London(UCL) was led by Professor B Cook.

This project focus on compositional analyses for binary code. Binaries are commonly studied by various groups such as the intelligence community, military organisations and security

engineers about their vulnerabilities. Binaries are usually hug thus relevant analysis and verification techniques must be scalable.

The scientific challenge in compositional reasoning is how to separate sophisticated interactions, prevent expensive operations, and derive procedure summaries that are compact.

1. **Program Verification Techniques for Understanding Security Properties of Software**
2. **Productive Security - Improving security compliance and productivity through measurement s**
3. **Partial order semantics for concurrent program verification**
4. [**Automating Separation Logic Reasoning**](http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/K040863/1)

<http://ieeexplore.ieee.org/document/1621009/?reload=true>