

Secure stepwise refinement: Beating the *Refinement Paradox*

Abstract

Stepwise refinement [Wirth 71] is the top-down presentation of a software system's functionality as a sequence of layers of increasing detail, beginning with the very abstract and ending with the very concrete, and with each layer an incremental "refinement" of the previous one. Making the separation between layers small means that each refinement step can be kept under conceptual control, in many cases even verified correct. Even though actually *developing* systems this way remains a theoretical ideal (sometime achieved), the refinement framework is in practice provides a framework for encouraging correct, accountable and even efficient code.

In spite of the conceptual and practical impact of Stepwise Refinement, it has not until recently incorporated a general treatment of security properties: this is due to the "Refinement Paradox," a fundamental interaction – even confusion – between abstraction and ignorance which appears to prevent refinement from preserving security properties.

The theme of this tutorial is to explore the relationship between refinement and security properties and to explain how a recent breakthrough has allowed standard refinement to be adjusted so that even security properties can be preserved. The tutorial will describe some of the underlying mathematical principles of refinement and security, will illustrate the paradox, and then will show how – by avoiding it – the refinement method can after all be applied to well-known security problems.

Outline

This tutorial describes recent advances in the stepwise-refinement method that incorporate the identification and preservation of security properties [Morgan 07]. The model and logic presented is inspired by, but abstracted from, the probabilistic refinement pioneered by Carroll Morgan and Annabelle McIver [McIver 05]. The objectives of this tutorial are to describe the underlying mathematics and resulting restrictions to standard refinement required in order to ensure that security properties are preserved. The methods will be illustrated by secure refinement laws and the verification of a non-trivial case study.

Detailed topics (3 hours estimated time, ie 1/2 day)

- Summary of a simple security-model for sequential programs, separating the data into "low"- and "high" security levels; an illustration of the "Refinement Paradox" (1/2 hour);

- A description of how to specify security properties in terms of what can be known of the contents of a file (1/2 hour);
- The definition, illustration and application of a secure refinement, ie a refinement which preserves all of the security properties (1/2 hour);
- General secure-refinement rules (1/2 hour);
- Case Studies: one or both (time permitting) of the well-known Dining Cryptographers and Oblivious Transfer protocols (1 hour).

Expected Audience

The tutorial will be most suited to people who have some experience with standard refinement techniques (eg action systems, the B method), or who have an interest in Formal Methods for security properties. No advanced mathematics will be assumed.

Speaker: A/Prof Annabelle McIver

Associate Professor Annabelle McIver has recently published (jointly) two texts in Formal Methods: one is a collection of articles by the internationally renowned membership of the IFIP Working Group 2.3 *Programming Methodology* [McIver 03]; the other is, to date, the only full research text on probabilistic program abstraction, refinement and proof [McIver 05]. She is the author of dozens of papers on Formal Methods ranging from highly theoretical (quantitative modal algebra) to extremely practical (automatic correctness verifiers for probabilistic systems). She holds degrees in mathematics from Cambridge and Oxford (UK), and has worked in industry. Currently she is based at Macquarie University in Sydney, and is a Fellow of Australia's National ICT centre where she applies mathematically based program-correctness techniques to the design and deployment of wireless sensor networks.

[Wirth 71] Program Development by Stepwise Refinement, Niklaus Wirth, Communications of the ACM, Vol 14, No. 4, April 1971.

[McIver 03] Programming Methodology, A McIver and C Morgan (editors), Monographs in Computer Science, Springer, 2003.

[McIver 05] Abstraction, Refinement and Proof for Probabilistic Systems, A McIver and C Morgan, Monographs in Computer Science, Springer, 2005.

[Morgan 07] The Shadow Knows: Refinement of Ignorance in sequential programs, C Morgan, in Mathematics of Program Construction (359--378) LNCS 4014, Springer 2007.