PhD Thesis Proposal

José Bacelar Almeida and Jorge Sousa Pinto
CCTC-UM

January 2008

1 Title

Domain-specific Program Verification Infrastructures

2 Context

This work is proposed in the context of the research projects *RESCUE, REliable and Safe Code execUtion for Embedded systems* funded by the Portuguese Foundation for the Science and Technology, FCT (*Fundação para a Ciência e a Tecnologia*), and *CACE, Computer-aided Cryptographic Engineering*, funded by the EC FP7. The DIUM/CCTC team in both projects consists of José Bacelar Almeida, Manuel Barbosa, Maria João Frade, and Jorge Sousa Pinto.

Both projects kick off in January 2008. In addition to team support, they will provide travel money, as well as fund other human resources (namely research assistant grants) to work in tasks closely associated with this PhD project.

The RESCUE project aims at providing innovative, efficient and expressive mechanisms for the secure implementation and execution of code, with an emphasis on problems posed by embedded systems. Innovative mechanisms are required to develop techniques that will allow embedded applications to be statically checked against safety policies, to self-adapt considering the availability of resources, and to perform software upgrades without human intervention.

The CACE workpackage relevant to the present proposal is concerned with constructing a toolbox for Secure Software Engineering. The goal is to bring the benefits of Formal Methods to Cryptography-minded Software Engineers, and in particular to apply the Design-by-contract approach to the development of secure software, as well as program analysis and verification techniques to a cryptographic domain-specific language.

3 Proof-carrying Code

Safety policies can give end users protection from a wide range of flaws in binary executables, including type errors, memory management errors, violations of resource bounds, access control, and information flow. *Proof Carrying Code* (PCC) is an enabling technology for the static and decentralized enforcement of complex and configurable security policies based on verifiable evidence, which has been successfully applied in various contexts.

PCC is a software execution platform that basically provides a clear and operational separation between the actors: who must convince and who must be convinced about security properties when sharing code. The code producer knows better how its code is built and how it behaves. Thus this actor is capable of ensuring that its code is safe. For such a task PCC advocates the production of verifiable evidence, in the form of a certificate (originally a formal proof), that the code does not contain flaws or can do no harm.

On the other hand, the code consumer (or receiver) knows better what is safe for it. It usually requires services from outside, in particular from code producers, and is capable of verifying if these services are compliant with its own security requirements. The certificate can be mechanically checked by the host; the producer need not be trusted at all, since a valid proof is incontrovertible evidence of safety.

Traditional PCC relies on the same formal methods as Program Verification in general (i.e. a Hoare-style Logic and theorem-proving support), but it has the significant advantage that safety properties are much easier to prove than functional properties.

4 Goals

The first main objective of the project is to bring advancements to the current state of the art in PCC technology concerning the specific setting of embedded systems. For this purpose, collaboration with a group of researchers (also participating in the RESCUE project) expert in Embedded Systems will be crucial.

The components of such a system, in which contributions are expected, include the definition of appropriate safety and security properties; the definition of an adequate *Programming Logic* and the design of the mechanisms that provide the construction of certificates. Outputs of this work will help setting up a complete PCC platform for static security enforcement mechanisms, and to provide a working deployment in embedded systems.

The design of such safety mechanisms will potentially give rise to a new embedded software paradigm. Safety certificates allow new execution schemes where, for instance, (i) a program can provide static evidence that it will not use unsafe operations or resources; and (ii) two applications can safely coexist in a embedded system.

The second main goal is to investigate at the theoretical level the benefits of the Design-by-contract approach, on one hand, and of the construction of a PCC architecture for Cryptography Engineering – a completely novel idea, which again requires a thorough study of the application area to identify the specific requirements. The first task involved is the examination of a domain-specific language from a critical perspective.