



Publishable Summary of the CACE project

Project number:	216499
Project acronym:	CACE
Project title:	Computer Aided Cryptography Engineering
Start date of the project:	January 1 st , 2008
Funding scheme:	FP7 ICT STREP

Date of the reference Annex I:	September 26 th , 2007
Deliverable:	Publishable Summary of the 1st period of the CACE Project (as part of the "1st Periodic Report of the CACE project")
Period covered:	01.01.2008 – 31.12.2008 (M01-M12)
WPs contributing to the deliverable	All
Due date	31 st December 2008 (M12)
Actual submission date	27 th February 2009

Responsible organisation:	Project coordinator: Technikon Forschungs- und Planungsgesellschaft mbH (TEC)
Tel.:	+43 4242 233 55
Fax:	+43 4242 233 55 77
E-mail:	coordination@cace-project.eu
Project website:	www.cace-project.eu

2 Publishable summary



Project Name: **CACE**
Grant Agreement: **216499**
Project Website:
Contact:

Start date: 1 Jan. 2008
Duration: 36 months
<http://www.cace-project.eu/coordination@cace-project.eu>

Mission of CACE

"To enable verifiable secure cryptographic software engineering to non-experts by developing a toolbox which automatically produces high-performance solutions from natural specifications?"

The CACE Project

Development of hardware devices and software products is facilitated by a design flow, and a set of tools (e.g., compilers and debuggers), which automate tasks normally performed by experienced and highly skilled developers. However, in both hardware and software examples the tools are generic since they seldom provide specific support for a particular domain. Within the CACE project a toolbox, that will support the specific domain of cryptographic software engineering, will be designed, developed and deployed.

Motivation

Ordinarily, development of cryptographic software is a huge challenge: security and trust is mission critical and modern applications processing sensitive data typically require the deployment of sophisticated cryptographic techniques. The proposed toolbox will allow non-experts to develop high-level cryptographic applications and business models by means of cryptography-aware high-level programming languages and compilers. The description of such applications in this way will allow automatic analysis and transformation of cryptographic software to detect security critical implementation failures, e.g. software and hardware based side-channel attacks when realising low-level cryptographic primitives and protocols. Ultimately, the end result will be better quality and more robust software at a much lower cost; this provides both a clear economic benefit to the European industry in the short term, and positions it better in dealing with any future roadblocks to ICT development in the longer term.

Objectives & Overall Strategy

The CACE project aims to target the lack of support currently offered to cryptographic software engineers. The central objective is the development of a toolbox that supports the production of high quality cryptographic software. The aim is that specific components within the toolbox will address specific software development problems and processes; combined use of the constituent tools is enabled by designed integration between their interfaces. A representative example use of the toolbox might be to develop an online voting system by a natural, high-level description of the system properties. The CACE toolbox would take this description and produce an efficient, executable implementation, which has verifiable security properties both at the semantic and physical levels.

The main technical objectives of CACE are therefore as follows:

- Development of a toolbox, which automates cryptographic tasks and therefore supports developers in implementing cryptographic schemes
- Automatic translation from natural specifications
- Automatic security awareness, analysis and correction
- Automatic optimization for diverse platforms

The CACE project is divided into three stages, which roughly correspond to the tasks of requirements analysis during the first year, development during the second year and evaluation during the third year.

Within the first project stage (M01-M12), all languages, compilers and static libraries are specified. Interfaces between WP deliverables are stabilized. Theoretical hurdles are identified and feasible solutions are outlined. Prototypes of lower-level compilers and shared library are released.

The second stage (M13-M24) covers the completion of the full implementations of lower-level compilers and shared libraries. Further the theoretical framework is completed and novel results are published. Prototype implementations of higher-level compilers and run-time environments are released (building on lower-level results).

Within the third and last stage (M25-M36), the optimization and extension of lower-level compilers and shared library is completed. Full implementation of higher-level compilers and run-time environments are released.

In order to ensure that these stages are completed in a timely manner, the overall execution of the project can be easily monitored by the following 6 major milestones, constituting central points in the course of the project across the technical work packages:

After 12 months:

- Milestone 1 – System analysis and specification completed.
- Milestone 2 – Proof-of-concept completed.

After 24 months:

- Milestone 3 – Full prototype of CACE toolbox available.
- Milestone 4 – Research results finalised and published.

After 32 months:

- Milestone 5 – System integration completed.

After 36 months:

- Milestone 6 – Final CACE toolbox available, evaluation published.

Figure 1 illustrates the 3 stages of the CACE project and the corresponding major milestones.

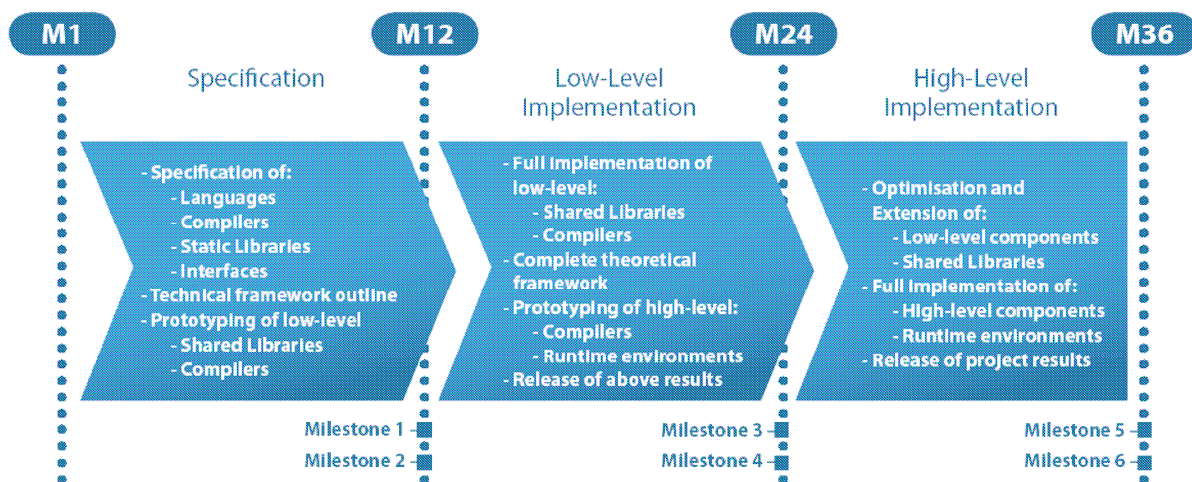


Figure 1: Stages of the CACE project

Technical Approach

The work plan of CACE includes six work packages. Work packages 1-4 deliver the tools, libraries and compilers for the toolbox. Formal verification and validation will be addressed in WP5. Work package 6 provides the organisational framework.

WP1: Automating Cryptographic Implementation

WP2: Accelerating Secure Networking

WP3: Bringing Proofs of Knowledge to Practice

WP4: Securing Distributed Management of Information

WP5: Formal Verification and Validation

WP6: Project Management, Dissemination and Standardisation

Description of the work done and the results in the first project year

The CACE project started in January 2008 and is going to run for 36 months. During the first project phase corresponding to the first project year the focus was put on the analyses and the specification of the requirements of the system and the creation of early prototypes of tools and libraries. All work packages started their work and produced altogether 14 deliverables spread throughout the first project year. At the beginning lots of effort was put into a successful launch of the project, a public project website and the internal IT communication infrastructure were provided as well as a dissemination plan for the entire project duration compiled.

For the technical work packages there were two general objectives for the first project year that were pursued and achieved across the work package boundaries. First, the main aim was to define the tools and the interfaces between the tools which constitute the CACE toolbox (for example languages, compilers and run-time libraries). Beyond this, the consortium aimed at producing early prototypes of tools and libraries as proof-of-concept and validation of the system analysis and specification results. Even though these prototypes are for internal use by project partners only, they are essential for the overall consistency of the project. This is particularly important for shared components such as run-time libraries, supporting the work of multiple work packages.

The progress achieved in all work-packages within the first project year is in line with the initial plan and can be summarized as follows:

Within **WP1** among other things deliverable D1.1, detailing the language definitions for CAO and qasm tools and enabling the implementation of shared run-time library components to commence, has been prepared. The significance of deliverable D1.1 is that with concrete and well considered language definitions for CAO and qasm, work on the associated compiler tools can accelerate. Based on early requirements analysis, a significant amount of work has gone into development of compiler/interpreter prototypes for CAO and qasm and the result is a solid source code framework which can be built upon.

WP2 produced 2 deliverables. In the first deliverable D2.1 in M06 a prototype implementation of the NaCl library with a C interface has been presented. It contains functionality for both secret-key and public-key cryptography. It does not yet contain networking functionality. The prototype is mostly written in C and the only supported interface language is C. An updated version of the prototype library has been implemented and presented in deliverable D2.2 which was due in M12 and which added networking elements to the prototype produced in D2.1. Further high speed implementations of different cryptosystems (AES, elliptic-curve Diffie-Hellman key exchange) have been written in qasm, demonstrating the usability of WP1 tools in WP2. Further a python wrapper was implemented and an independently developed python NaCl was studied. Also a C++ version of NaCl has been prepared. So far hash functions and AES are implemented.

The work done by **WP3** during the first year was following three major lines of research. One was focusing on reviewing and unifying the existing theory of efficient zero-knowledge proofs of knowledge (ZK-POK) (this work corresponds to Deliverable D3.1). The second line of work was to investigate and implement a first prototype of a compiler to generate implementations of ZK-POK (this work corresponds to Deliverables D3.2 and D3.3). The third line of work concerned the investigation of formal verification techniques for assuring the correctness of the compiler suite to be developed (this work corresponds to deliverables which are due in a later phase of the project).

WP4 has produced an overview of practical applications of secure multiparty computation, and has used this to define the essential properties that protocols should have to support the applications. This will be used to guide the protocol design work in the rest of the project. A domain specific language for secure multiparty computation has been specified, this specification will be the point of departure for the implementation in the coming periods. Further a virtual machine for multiparty computation has been designed, and a prototype has been implemented. Design and implementations of a number of cryptographic protocols have been done, these will be used for supporting the virtual machine. In summary, 4 deliverables have been completed.

Within **WP5** the milestone M5.1 - End-user and partner meeting for security policy identification – has successfully been organised in Porto in July 2008, which allowed the collection of an exhaustive list of types of security requirements needed to be addressed from all partners in the project. Further deliverable D5.1 has been completed within schedule. It contains a taxonomy of security policies that are relevant for the CACE project, and identifies to which CACE tool-box component each policy may apply. This document is an important reference for future work in WP5 (and other work packages) as

it delimits the design goals for the formal verification functionality to be developed in years 2 and 3. Additionally a technical report associated with M5.2 was finalized within schedule. It contains the specifications of the formal verification and validation tools that will be developed in WP5. This was an important achievement because the planned progress of WP5 relied on the fact that this could be done by the end of year 1.

As stated here, all the relevant overall project objectives have been successfully achieved and finished, the project made excellent progress and is consistent with the original planning. This paves the way for a successful second project year without any deviations from the original schedule.

The CACE consortium

The EC FP7 project CACE brings together leading companies and academic institutions in the area of cryptography. Together they represent a vertically integrated consortium, with knowledge stretching from the basic research (academic partners) to the design and marketing of products (industrial partners and SMEs). To foster the cooperation with the industry, several experts support the project consortium as Advisory Board members. As a lot of basic research and basic input on cryptography is needed for the CACE project, seven European universities as well as one Asian university participate in this project. Furthermore, a global industrial player and three SMEs contribute with their expertise and knowledge to this project. The 12 project partners are located in eight European countries (Austria, Denmark, Finland, Germany, United Kingdom, Netherlands, Portugal, and Switzerland) and one Asian country (Israel).



Figure 2: The CACE consortium at the Kick-off meeting in Bochum in February 2008

CACE Disclaimer

All public information will be marked with the following CACE project disclaimer:

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The user uses the information at its sole risk and liability.