Load time on-card application certification for Java Card:

The Security-by-Contract scheme for multi-application Java Card cards

Olga Gadyatskaya, Fabio Massacci (University of Trento)

SUMMARY

We enable multi-application Java Card-based secure elements by adding the *Security-by-Contract* (SxC) framework to the Java Card Run-time Environment. The SxC framework performs the load time application certification executed by the card itself. The framework enables applets to define their own service access control policies outside the functional code.

During the loading process the framework checks that the applet invokes in its code only the services it was authorized to invoke by the respective services owners. Therefore the SxC framework protects the application services while enabling possibility of independent asynchronous evolution of applets coming from different providers. The proof-of-concept implementation demonstrates feasibility of the approach.

KEY FEATURES

- **Practical service access control** – Functional code of applets is no longer interleaved with the access control code. The security policy of applets is delivered within Contract Custom components of the CAP files.
- **Flexible application policy updates** – An application policy can be updated without complete reinstallation of the applet.
- **Compatibility with JCRE 2.x platforms** – The framework is directly integrated with the JCRE Loader and Installer components. The SxC workflow is interleaved with the standard Java Card application development and deployment processes.
- **Non-invasive implementation** – Only several functions have to be added to the Installer and the Loader, no changes are applied to the virtual machine implementation or the firewall
- **Small memory footprint** – The SxC framework on-device non-volatile memory footprint is only 8KB. No RAM allocation is required, the prototype works with a 256B auxiliary temporary buffer.
- **The Developer SxC prototype** – SxC framework also exists in the developer version that runs on a PC, it can be used by the developers to practice the certification process.

IMPLEMENTATION HIGHLIGHTS

- The SxC framework consists of the native part (written in C, integrated with the Loader) and the Java Card part (written in Java Card, integrated with the Installer)
- The native part consists of two components: the SxCInstaller (serves as an interface with the Installer, includes the PolicyChecker component) and the Claim Checker. Both components use the Loader functions to access the CAP file components.
- The Java Card part, called the PolicyStore, maintains the security policy of the card across card sessions.
- The security policy format is highly optimized in order to minimize the EEPROM consumption. The bit vector operations are used to manipulate with the policy.
- Application contracts, containing the security policy and the details of service invocations, are developed with the CAPModifier tool, which can then append the contract to a standard CAP file converted by any available Java Card development kit. The CAP files containing contracts can be delivered on the card following the standard loading protocol.

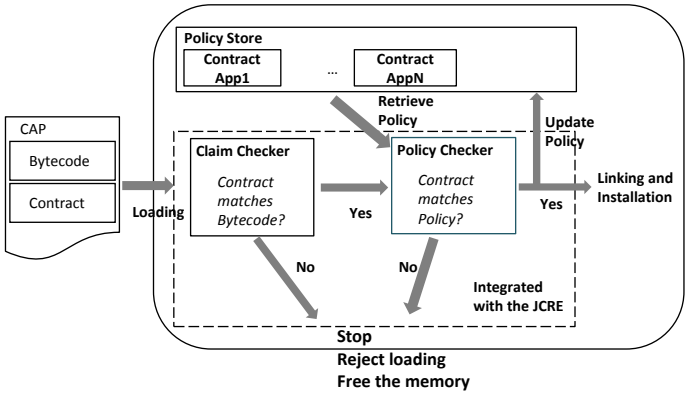

Figure 1. The SxC workflow for application loading

ENABLING EVOLUTION

With the advent of the new NFC-based ecosystem and the apps' marketplaces smart handsets are providing increasingly sensitive services that can be updated over the air in a dynamic fashion. The deployment of Java-enabled USIM/SIM cards which use the GlobalPlatform card technology have further enabled OTA application downloads for mobile networks. For security reasons these services are usually hosted together on a secure element.

A common assumption is that only few and limited applications will be loaded on the secure element but this is no longer the case: the usage of trusted elements evolves quickly following the trends in the smartphone markets. And if from a security perspective it is important that applications are confined, but from a business perspective we would like them to talk to each other within the secure element. In the same time it is crucial that each application provider is able to update the deployed applet and adjust its security policy independently from other stakeholders.

We target Java Card-based secure elements, such as (U)SIM cards. A (U)SIM card is already deployed secure element infrastructure on all mobile devices, while other secure element technologies can vary from device to device. In the same time Java Card enables deployment and execution of fully-fledged applications in a secure environment, while some secure element technologies are very restrictive and can be used only for storing sensitive data, like cryptographic keys of every provider.

The current setting of the Java Card technology deployed on the cards in the field (specification 2.x) allows the applications to interact only through specific Shareable interface methods, also called services. The application can control access to their services directly in the functional code by using some Java Card API to retrieve the caller application identifier (AID) and matching it with some list of trusted or forbidden clients in the code.

This solution essentially interleaves the security code with the functional code, what is a known source of bugs. Moreover, this hinders the possibility of applications to update their own security policies, because the only way to update the policies in the current setting is to reinstall the CAP files.

The SxC framework achieves the same security (the access control is defined per service and authorizations are given per CAP file), but it provides more flexibility, because the policies now can be updated without reinstallation of the functional code. Our approach makes a step toward open secure elements where each service provider can upload applications and maintain them independently from the other stakeholders.

We have developed the proof-of-concept embeddable SxC prototype, which performs the loading time application code certification. The on-card application certification for CAP file loading is summarized in Figure1. The main idea of the SxC scheme is that each application is loaded together with its contract. A contract contains the security policy of the application and the details on provided and called services. The Claim Checker component of the SxC framework verifies that the provided and called services details are faithful. Then the Policy Checker component ensures that the application calls only the services it is authorized to call, and the other applications on the card invoke only the services of this application they were allowed to call in the security policy. If both checks are successful the application security policy is added to the global on-card Policy Store and the CAP file can be linked. Otherwise the loading process is stopped and the CAP file is discharged.

The SxC proof-of-concept implementation was evaluated by a smart card vendor and integrated with an in-use Infineon smart card integrated circuit. The prototype is able to process applets of sizeable complexity. This implementation has demonstrated the potential of the approach.

For demonstration purposes we have also developed a prototype implementation that can be run on the PC simulator. This implementation allows the developers to practice the Security-by-Contract approach before installing the framework on a real card.

More information at
http://disi.unitn.it/~gadyatskaya/sxc.html

UNIVERSITÀ DEGLI STUDI DI TRENTO

secure CHANGE