SCMS CV Pilots Documentation

Security Credential Management System Proof–of–Concept Implementation

EE Requirements and Specifications Supporting SCMS Software Release 1.2.2

Made Available to the United States Department of Transportation

National Highway Traffic Safety Administration (NHTSA)

November 15, 2016

In Response to Cooperative Agreement Number

DTNH22-14-H-00449/0003

Notice and Disclaimer

This material is based upon work supported by the U.S. Department of Transportation under Cooperative Agreement No. DTNH22-14-H-00449/0003.

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the Author(s) and do not necessarily reflect the view of the U.S. Department of Transportation.

Table of Contents

Introduction

The Security Credential Management System (SCMS) Proof-of-Concept (POC) Implementation Project (SCMS POC Project) is being conducted by the Crash Avoidance Metrics Partners LLC (CAMP LLC) Vehicle Safety Communications 5 (VSC5) Consortium. Members of the Consortium are Ford Motor Company, General Motors LLC., Honda R&D Americas, Inc., Hyundai-Kia America Technical Center, Inc., Mazda, Nissan Technical Center North America, Inc., and Volkswagen Group of America. The goal of the SCMS POC design is to provide security services to support Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications at current production levels of passenger vehicles (up to 17 million annually) for the first year of deployment. An important goal of the SCMS POC system is to provide a flexible architecture that is capable of scaling to support larger numbers of V2V and V2I devices in the years following initial deployment. It is also anticipated that the SCMS POC design will provide both a stable platform and a research platform to support the USDOT and industry research needs prior to deployment. The work is sponsored by the National Highway Traffic Safety Administration (NHTSA) through Cooperative Agreement DTNH22-14-H-00449/0003.

Work in Task 4 of the project focuses on the design of the SCMS core components and protocols. Four software releases are planned during the course of the project. This document presents the requirements and specifications for the **SCMS POC System Release 1.2** from the perspective of an **End Entity (EE)**. This document is a work-in-progress. Future refinements and revisions to the requirements and specifications are anticipated as SCMS refinement is an ongoing task across multiple projects.

Introduction for EE Developers

The following paragraph will guide you as an EE developer through this documentation highlighting requirements and API documentation in the order of an EE's lifecycle. If you implement your EE software following this guide, you should have a device at the end that is able to communicate with the SCMS throughout the whole lifecycle.

1. First of all you need a Secure Environment for Device Enrollment where initialization and bootstrapping of your device will be executed
2. You need to have a device that applies to the requirements and descriptions laid out in Hardware, Software and OS Security Requirements
3. You need to have a True Random Number Generator
4. Your device needs to support in either hardware or software Approved Cryptographic Algorithms
5. You need to have an HTTP client that is able to communicate securely (HTTPS) to the SCMS as described in EE-RA Communications - General Guidance and EE-SCMS Core Communication Requirements
6. You need to know which Certificate Types you need to have on your device, which depends on the SCMS PoC Supported V2X Applications that you want to run on your device
7. The EE lifecycle starts with Use Case 2: OBE Bootstrapping (Manual), respectively Use Case 12: RSE Bootstrapping (Manual) depending on your EE type (OBE vs. RSE). Currently both processes are exactly the same.
8. Based on the EE type you are developing, you then create and send one of the following requests. All devices should always check for a new local certificate chain file (API: RA - Download Local Certificate Chain File) and a new local policy file (API: RA - Download Local Policy File) before sending subsequent request. All requests in this step #8 should be sent within the same HTTPS session.
    a. Pseudonym Certificates:
        i. Following the process in Use Case 3: OBE Pseudonym Certificates Provisioning, your OBE should create a pseudonym certificate batch request as described in Step 3.1: Request for Pseudonym Certificates and send it to the RA API as documented in RA - Request Pseudonym Certificate Batch Provisioning. Your OBE needs to create the butterfly seed pairs as described in SCP1: Butterfly Keys. Your OBE will get a response from RA with an *URL* and a *download time*.
        ii. Once your OBE's clock reaches *download time*, your OBE can download the initial pseudonym certificate batch at *URL* following the process in Step 3.3: Initial Download of Pseudonym Certificates using the RA API as documented in RA - Download Pseudonym Certificate Batch and the .info file using RA's API documented in RA - Download .info File.
    b. Application Certificate:
        i. Following the process in Use Case 13: RSE Application Certificate Provisioning, your RSE should create an application certificate request as described in Step 13.1: Request RSE Application Certificate and send it to the RA API as documented in RA - Request Application Certificate Provisioning. Your RSE will get a response from the RA with an *URL* and a *download time*.
        ii. Once your RSE's clock reaches *download time*, your RSE can download the application certificate at *URL* following the process in Step 13.3: Download RSE Application Certificate using the RA API as documented in RA - Download Application Certificate.
    c. OBE Identification Certificate:
        i. Following the process in Use Case 19: OBE Identification Certificate Provisioning, your OBE should create an identification certificate request as described in Use Case 19: OBE Identification Certificate Provisioning and send it to the RA API as documented in RA - Request Identification Certificate Provisioning. Your OBE will get a response from RA with an *URL* and a *download time*.

    ii. Once your OBE's clock reaches *download time*, your OBE can download the identification certificate at *URL* following the process in Step 19.3: Initial Download of OBE Identification Certificates using the RA API as documented in RA - Download Identification Certificate and the .info file using RA's API documented in RA - Download .info File.
9. Depending on the certificate type, the SCMS constantly pre-generates them and your EE can download top-offs like this:
    a. Pseudonym Certificates: Whenever it suits your pseudonym certificate download strategy at a point of time that is after the time given in the .info file, follow the process described in Step 3.5: Top-off Pseudonym Certificates using RA's API documented in RA - Download Pseudonym Certificate Batch to download additional pseudonym certificates.
    b. Identification Certificate: Whenever it suits your identification certificate download strategy at a point of time that is after the time given in the .info file, follow the process described in Step 19.5: Top-off OBE Identification Certificates using RA's API documented in RA - Download Identification Certificate to download the next identification certificate.
10. Your EE should download the latest CRL as often as possible but no later than once a week using the process described in Use Case 6: CRL Download using the API documented in MA - Download CRL.
11. Your EE must verify incoming messages. Part of the verification is to check if the senders certificate was revoked following the process described in Step 8.4: OBE CRL Check, respectively Step 16.4: RSE CRL Check, as well as if a CA certificate in their certificate chain was revoked.
12. Report misbehavior: This is still TBD and will be supported with SCMS Release 2
13. Re-enroll: This is still TBD and will be supported with SCMS Release 2