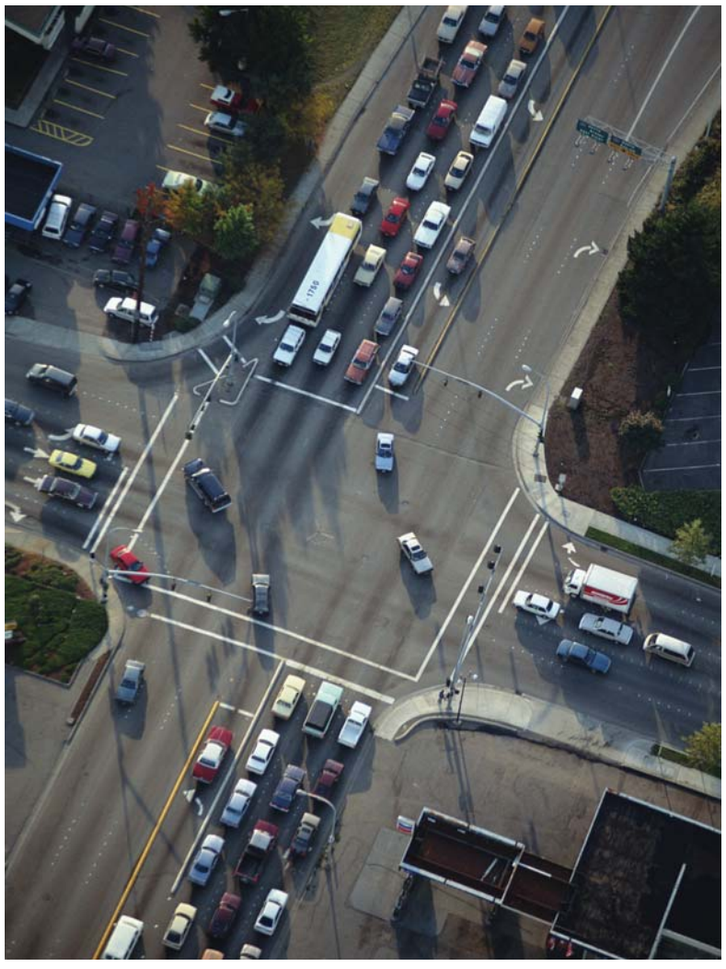
FOAM-filled

The security systems and applications currently being investigated by the FOAM sub-project of CVIS will lead to the safe and secure download of the services which will populate Europe's future cooperative transport networks. Mick Baggen (Technolution), Andreas Schmid of (PTV) and Zeljko Jeftic & Peter Christ (ERTICO) explain how it all works



The CVIS architecture's common building blocks can be deployed in vehicle, roadside or control centre systems. This ensures the core functionality and a security component will be present at any given node

In August this year the CVIS (Cooperative Vehicle-Infrastructure Systems) project revealed its vision of an architecture for a cooperative infrastructure where vehicles and infrastructure communicate and cooperate with one another in order to enhance on-road safety and efficiency. CVIS is a large-scale research project funded by the European Commission intended to establish European technology leadership in cooperative systems.

The CVIS project's architecture will support new ways of communication and cooperation which offer a breakthrough for ITS deployments. Building upon results from the GST (Global System for Telematics) project and adapting the results from the SeVeCOM (Secure Vehicle Communications) project, CVIS has further extended its overall architecture to include roadside systems and the additional needs of key stakeholders inside a cooperative system.

This cooperative system relies heavily on wireless communication and a distributed architecture where security components need to be individually distributed. The safe and secure download of services is the responsibility of FOAM, a sub-project of CVIS.

Security in action

The CVIS architecture is based on functional building blocks which can be deployed to any CVIS node. Functional blocks can be deployed in vehicles, roadside or control centre systems during system runtime using JAVA/OSGi-based mechanisms. Common functional blocks ensure the core functionality of the systems and a security component will be present at any given node.

To illustrate this, imagine a 'cooperative vehicle' crossing a national border in Europe. In his or her home country, the driver has a contract with a service which supports dynamic routing and requires participating vehicles to provide floating probe data.

A similar service is available in the neighbouring country and CVIS nodes announce this to the vehicle system as it enters. The driver has a 'Europe' contract and service in the region is part of the European service offer (although from a different provider), so the driver seeks to use it. The local version uses different software and interfaces, however the remote management function of CVIS allows the download of the appropriate software.

Security architecture elements have to ensure that: the software is not malicious to the vehicle system; the vehicle may actually use the service; and the driver doesn't need to re-subscribe manually to the local service as it is part of an existing contract.

The driver is automatically provided with the new service, however whether at home or abroad further concerns must be addressed, that: no-one can eavesdrop on the communication between the service provider and the vehicle or manipulate the communication content; other applications in the vehicle must not use data from this service without permission; and no individual vehicle (or even driver) tracking is facilitated through the application. The application itself must authenticate to make sure that no malicious participant can send manipulated or wrong information.

To tackle all of these issues CVIS has adopted the following elements in its architecture: authorisation and authentication (distributed authorisations, and authentication services which serve multiple business stakeholders based on single sign-on and federated identities capabilities); secure communication (including over multiple nodes with different encryption mechanisms) and a secure module providing trusted execution platforms; and an identity manager which provides privacy-enhancing measures.

For drivers on longer journeys, the automated provision of local services is possible

Remote management

A core CVIS architecture feature is the ability to manage remotely the lifecycle of software on a client system. The different lifecycles of hardware and applications and the flexibility to offer dynamically different or new services in different regions have driven this need. It means that manufacturers of vehicle or roadside systems need a mechanism to control what is happening on platforms for which they are responsible.

CVIS assigns each client to a Host Management Centre (HMC). Through these HMCs, operators can fully control which software can or cannot be downloaded to a client. The CVIS architecture enables the rules for download to vary from very strict to fairly relaxed. Operators can also check the client system state and, for instance, carry out dependency checks between software components or check if the right hardware and data resources for an application are available.

Authentication and authorisation

Authentication and authorisation are essential mechanisms of today's communication systems and Internet services. Almost every Internet portal site offers special services for registered users. A single user can have multiple username/password combinations for numerous Internet portals and this has led to the development of a new service called identity federation. This service allows use of a single username/password combination for several service providers. Additionally, the services can be used seamlessly without having to authenticate multiple times. The use of connected services and portals is therefore much more convenient for the end user.

The idea of identity federation works as follows. Several service providers collaborate and make an agreement or contract of collaboration. Based on this, a so-called circle of trust is formed. All partners within this circle trust each other and allow seamless transfer of service consumption.

A potential user of some or all of the providers within the trust environment has to sign up at the federation entity, which in the CVIS context is the HMC. Within the HMC, all user IDs and subscriptions are collected and a federated user profile is generated. After subscription, the HMC informs the respective service provider of the new customer and its preferences. The provider responds appropriately. This newly generated ID is federated with the existing profile of the user at the HMC. In the cooperative systems world, such a circle of trust could be established between a vehicle manufacturer, service provider and road operators.

In order for a circle of trust to be functional, distributed authentication is required. Only one authentication operation is needed to enter the circle of trust so the authentication operation has to be distributed to satisfy all needs within the system.

The service platform provides the following mechanisms for distributed authentication and authorisation: an authentication broker which facilitates the use of single sign-on services provided within the CVIS system; and an authorisation broker which provides users with access to distributed services.

Secure service download

The CVIS architecture for secure communication is based around the creation of secure communication 'tunnels'. To realise these, the project introduces the concept of the secure module. The secure module is a combination of hardware and software components used for tamper-evident operations such as on cryptographic keys or sensitive data. It can provide the necessary cryptographic functionalities for secret and public key operations within CVIS. Depending on a node's actual implementation it can include a number of functions.

Each CVIS node includes its own security module and relies on it to determine the authenticity of information and also to encrypt and decrypt information which is to be kept confidential.

For secure communication to take place between two nodes, a secure communication tunnel is established between their security modules. Depending on the security policy of the nodes involved, the security modules validate incoming messages and authenticate the outgoing. If a node encrypts its outgoing messages, the receiving node's security module will have to decrypt. The secure communications tunnel can protect the integrity of the information sent through the tunnel and also the information's confidentiality. The initialisation of the secure communications tunnel takes place when the client node and server node discover each other.

For each secure communications session, the relevant security policy may require any of the following security levels: insecure communications (where the node requires no specific cryptographic mechanisms for protecting the authenticity or confidentiality of the information transferred; an example might be information from road sites on the distance to the next service point, parking area and so on); authenticated (where the node which uses a secure communication service requires that the integrity of the information exchanged is protected and that the sender of the information must be authenticated; roadside units, for instance, will require authentication from vehicles and vice versa to exchange floating car data); confidential (where the node which uses a secure communication service requires that the confidentiality of the information exchanged must be protected; information on vehicle data and sensors will need a certain level of confidentiality); and secure (where the node which uses a secure communication service requires that both the confidentiality and integrity of the information exchanged through the service must be protected).

The concept of security modules and a secure tunnel can be combined for communications over multiple nodes with different encryption mechanisms. A translation mechanism is then used.

The secure communication engine takes care of all communication-related tasks. It chooses the properties for the communication channel based on the security policy set for the service or application.

Identity manager

CVIS has extended its security architecture to support privacy-enhancing technologies. Currently, members of the SeVeCOM project are working on a concept of using pseudonyms to protect privacy which CVIS will incorporate into its architecture. A new identity manager component is responsible for creating the desired level of anonymity.

An identity manager is added as an extra security component in FOAM. The identity manager is responsible for storing and leasing pseudonyms to client application. A client application is used for loading new pseudonyms into the identity manager. The pseudonym client application will periodically contact a pseudonym provider to fetch new pseudonym certificates and private keys. The pseudonym provider will administrate the relation between the generated pseudonyms and the authentication credentials of the client node for exceptional circumstances where the real identity needs to be resolved from a given pseudonym certificate. Currently the identity manager infrastructure is also under investigation within SeVeCOM. ■

www.cvisproject.org