Access Control for Automotive Systems using VISSv2 Protocol

José J. Sánchez-Gómez, Isaac Agudo
NICS Lab, University of Malaga

December 2, 2022

1 Introduction

Vehicle System Specification (VSS) is an specification defined by W3C to describe the data schema of a vehicle. VSS addresses the data of the vehicle using a hierarchical tree like taxonomy with nodes defined from the root of the tree. One example of this addressing is the state of a window of the vehicle: "Vehicle.Cabin.Door.Row2.Right.Window.IsOpen" or the actual speed of the vehicle: "Vehicle.Speed".

Vehicle Information Service Specification version 2 (VISSv2) is a service that exposes the nodes defined in VSS. These nodes are exposed within the vehicle's network or to the Internet using WebSockets, MQTT or HTTP. Read, Update and Subscribe requests for the data are supported. Message syntax and some pre-defined messages can be found in the protocol specification.

The use of a well-defined Access Control Model reduces the risk of a successful attack coming from an unauthorized intruder. Different actors and policies are implemented in order to manage the Access Control of the data in the vehicle and restrict the access to actuators and data in the Vehicle.

2 Access Control Architectures

2.1 Fully Independent Actors. Token Based.

Three different actors interact with the client to implement a Token-Based Access Control Schema. The different phases of the Access Control are delegated to each one of these actors:

- **Access Grant Token Server.** This actor must authenticate the client, which is described by a context. The client context includes an application, user and device role. These roles allow to classify each of the possible types of role in the ecosystem.
  The client must attest its context by sending a set of proofs (one for each role), which must be included in the request.
  After verifying that the client context is valid, the AGTS (Access Grant Token Server) issues an (AGT) Access Grant Token signed using its private key.

- **Access Token Server.** The authorization is performed by this actor.
  The client must send the AGT issued by the AGTS accompanied with the set of signals or the purpose the access to is being requested.
  The ATS (Access Token Server) must know the AGTS signature in order to validate the AGT received. AGT expiration time must also be checked.
  If the received AGT is valid, the ATS must check if a client with the context included in the token can access the signal(s) that are being requested.
  After checking that the signals can be accessed, the ATS generates a signed AT (Access Token) that contains the signal(s) that can be accessed.

- **VISS Server.** This server has access to all the signals in the vehicle.
  VISS Server must receive a request containing an AT and a set of signals to access. The server must verify the AT signature and check if the signal(s) and the action requested is allowed by the AT.

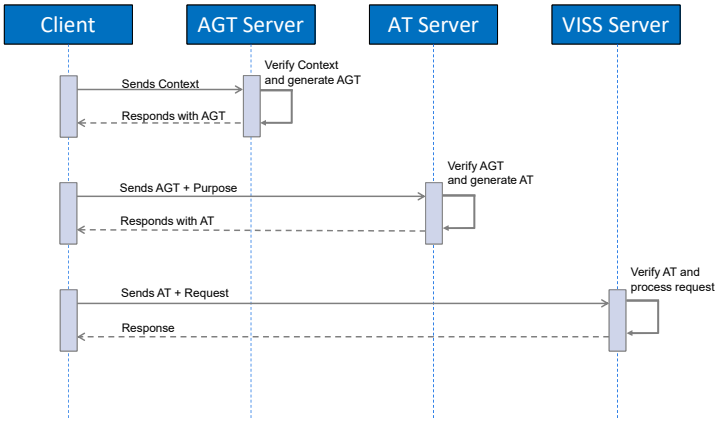

Figure 1: Fully Independent Access Control Authorization Sequence. Note that all requests are supposed to be valid.

The four elements shown in Fgiure 1 are run independently and could even be deployed in different trust domains. That forces them to share some kind of cryptographic material in order to authenticate requests. In case they use Symmetric cryptography, i.e. HMAC functions, AGT Server and AT server need to share a secret key and AT and VISS server need to share a different secret key. If case they use asymmetric cryptography, i.e. Digital signatures, they need to share their public keys, which makes the setup much simpler and scalable.

2.2 Semi-Independent Access Control

Access Token Server and VISS Server can establish a secure connection. The process will be similar to the one described in figure 1. Given the advantage of the secure connection between these two ends, VISS Server could delegate the Access Token checking procedure to the AT Server.
The VISS server and the AT server don't need to share any cryptographic material in this case, once they have already established the secure channel.

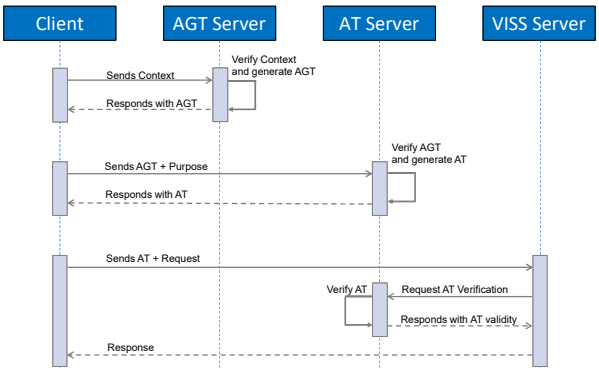

Figure 2: VISS Server delegates the AT verification to the Access Token Server. Note that all requests are supposed to be valid.

Using this scheme, computing load on VISS Server will be reduced, since all the cryptographic tasks performed to verify the signature of the Access Token or checking the possible client certificates would be delegated to the ATS.

The ATS can either verify the request made to the VISS Server or only the Access Token. In the first case, the VISS Server must check the authorization using the purpose contained in the AT, delegating the signature and token verification to the ATS.

2.3 Semi-Independent Access Control using opaque tokens

The schema described in figure 2 allows to delegate the AT validation to the ATS. In case the VISS Server permanently delegates the validation to the ATS, opaque tokens can be used in order to authorize the client, since the syntax of the token must be known only by the ATS.
The Access Token Server will make use of token containing UUIDs or similar claims instead of Access Tokens. These UUIDs will be cached or stored by the ATS, including the set of signals or the purpose requested by the client, allowing to validate these requests during an established caching time that will be decided by the ATS.

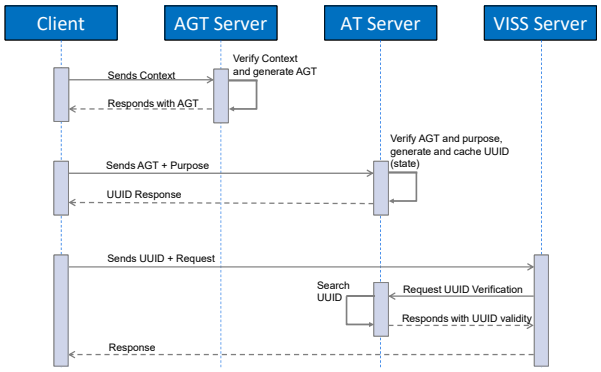

Figure 3: VISS Server delegates the verification of the Authorization to the VISS Server. Valid requests are supposed in this diagram.

In both this case and the Semi-Independent Access control, a set of signals is requested to the VISS Server. In this case, the verification needed to check if the client can access the set of signals must be performed by the ATS, since the structure of the opaque token can not be verified by the VISS Server.

2.4 Access Token Server as Policy Enforcement Point

In order to minimize the quantity of requests and the relative complexity in the environment, the Access Token Server can be suppressed by including a Policy Enforcement Point (PEP). The PEP and VISS Server can either be running in the same machine or in different environments as long as a secure and persistent connection can be established between them. Low latency and enough bandwidth must be ensured in order to allow a fast verification
VISS Server will ask to the PEP if a client has access to a set of signals. The PEP will receive an Access Grant Token and the set of signals requested by the client. After verifying that the client can access that set of signals using the context in the AGT and the validity of the AGT, it will return a response addressing the signals that the client can access.
Given that the AGT will be repeatedly used when requesting data, the client might hold a cryptographic keypair, being the AGT linked to the client public key, thus being a Long-Term AGT (LT-AGT). When the client uses a LT-AGT, it must be accompanied by a proof of possession of the key linked with the LT-AGT. This ensures protection if this token is stolen by an eavesdropper and allowing it to have a longer expiration time.
To allow faster data interchanges, the PEP might use a stateful or connection based strategy to store information about the AGTs received and the signals that can be accessed using them. If this strategy is adopted by the PEP, the client might not need to repeatedly send the AGT during a given time interval. Alternatively, the PEP could store the AGTs associated with a client key to allow the client to only send PoPs in order to authenticate itself.

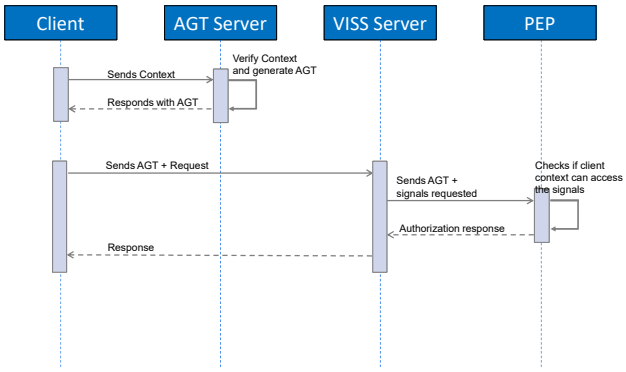

Figure 4: Access Control using a PEP that receives the AGT and signals requested. Valid requests are supposed in this diagram.

This strategy ensures a much-more efficient use of the communication channel and allows the client to request data without having to care about requesting and storing Access Tokens. In contrast, using AGTs accompanied by Proof of Possessions is a good practice since AGTs are repeatedly used to request data.

2.4.1 Without Access Grant Token Server

In some cases, the Access Grant Token might not be generated by the AGTS. In these cases, the AG-Token must be generated and provided to the client in a secure environment. Although the use of an AGT is described in this section, other type of bearer authentication might be used instead of an AGT.

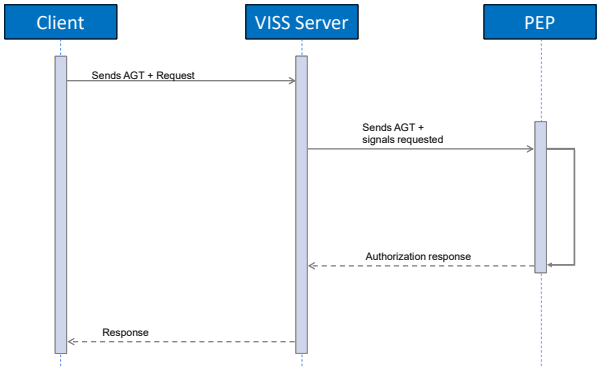

Figure 5: The method used to obtain the Access Grant Token is out of the scope in this diagram. The AGT could be replaced by any other type of bearer.

This token could be provided to the client by the manufacturer of the vehicle, the fleet manager or a secure device dedicated to generate these kind of tokens.

A hard-coded and permanent Access Grant Token might be used. In this case, the AGT must be linked to the client Public Key and when sent, it must be always accompanied by single-use and recent proof of possession of that key.

2.5 Preinstalled in-Vehicle Access Grant Token supporting delegation

There might be some cases in which the full control of the vehicle is delegated to the owner of the vehicle or someone having physical access to the vehicle. A *master Access Grant Token* linked to a public key can be pre-installed in the vehicle and have an expiration time enough to be used during the lifetime of the vehicle. Therefore, it must be ensured that this AGT and its linked Key-Pair is stored in a highly secure environment, and used in a trusted internal network. Although this is out of scope, a good practice could be storing the KeyPair in the physical key of the vehicle, which should have cryptographic capabilities in order to perform signatures when needed.

All the Servers and Actors implied in the Access Control Schema must be installed in the vehicle and these must depend on the master AGT. This AGT will be used by the OEM systems of the vehicle allowing them to access all required data in the vehicle.

An hybrid ecosystem can be implemented. It will consist on a combination of all the described alternatives: Access Grant Tokens can be used to request data using the PEP and Access Tokens can be used via the Access Token Server.

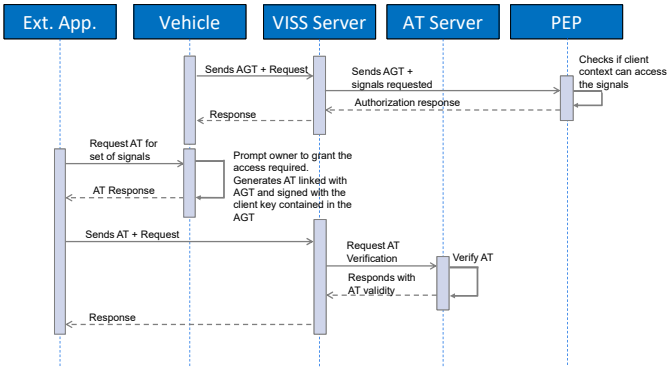

Figure 6: Vehicle holds the master Access Grant Token. Access Tokens can be generated for clients after authorization of the user of the vehicle.

The main advantage of this schema is that it gives full control of all the signals in the vehicle to the owner/driver.The vehicle will allow external connections with other devices, such as smartphones or diagnostic devices. Similarly, the vehicle will have an internal untrusted network allowing applications installed and devices directly connected to the vehicle to access the data of the network. The master AGT must never be sent through these two networks as mentioned earlier.

Two interesting use cases can be mentioned. In the first one, the vehicle is in a mechanical workshop, therefore, the mechanic should have access to a set of signals based in its context. In order to access these signals, the mechanic can request an AGT containing its context. This AGT will be generated by the Vehicle after prompting the owner of the vehicle for the duration of this token and requesting permission to generate it.

In the second case, the owner of the vehicle might want to install an application that will be able to remotely track the vehicle. This application will connect to the vehicle using a network that is external to the vehicle. After connecting to the vehicle, this application will request for an Access

Token containing a set of signals that it will use. Therefore, the UI of the vehicle will prompt, notifying about this request and prompting the signals whose access is requested, similarly to the app permissions schema used in Android Smartphones, where each one of the permissions can be individually given or not to an application. In this case, the owner might want to give access to the location, accelerometer and horn of the vehicle, in order to act as an alarm system. Nevertheless, although the application could offer the possibility of opening the vehicle remotely, this might not be wanted by the client, so that signal access request can be denied, issuing an AT including only the signals allowed by the person in the vehicle. These Access Tokens would be able to be revoked whenever the client requests for it, and should be renewed by the application in order to not prompting the user again.

3 Brief Evaluation of the Control Models

There are different alternatives depending on the interactions between the different actors of the Access Control Model and the tokens used in each one of the stages of the process.

3.1 Depending on the Interactions in the Ecosystem

Since distinct alternatives are described in this document, there are different primitives that must be followed when deciding which kind of interactions and connections between the different actors of the access control ecosystem are meant to be established. Although the client should always be able to communicate with all of the servers involved in the Access Control, these do not have to be able to communicate to each others.

3.1.1 Independent Schema

Using a fully independent schema allows the actors of the architecture to work in different environments, disregarding the rest of the ecosystem, since the interaction with the rest of the actors in the access control architecture are not performed directly.

Interactions between different actors in the access control architecture are performed using JWTs. These JWTs must be signed using Private Key Cryptography in order to allow the rest of the ecosystem to verify the issuer of the token and allow a secure signature (using a symmetric key in order to do this is not recommendable due to possible key leaks in the actors that must check the token).

Since no connection with the issuer of the token is avaliable, the syntax of the tokens and claims used must be shared between the actors of the ecosystem that indirectly interact via the client. These tokens must be validated using these parameters by the receivers of the tokens.

Access Grant Tokens could be generated when connected with the AGTS and then safely stored to use them when required, then, since no connection with the AGTS is required unless an AGT is being requested, as long as the rest of the ecosystem is avaliable and an AGT is owned by the client, data requests can be correctly performed.

Something similar happens with Access Tokens. The main difference is the short duration of these tokens, which makes them need to be frequently (re)generated. The client must therefore request these tokens with enough time to receive a new AT in case the connectivity with the ATS is lost.

This schema has some disadvantages, like the requirement of a secure storage in the client, in order to store the received bearers. No intruder with physical or local access to the client should be able to obtain the tokens stored, specially AGTs. To solve this, the client might be able to run cryptographic primitives and securely hold a Public-Private KeyPair. Tokens issued for a client being able to perform this task can be linked to the client public key, allowing them to have a longer expiry time with the condition of being accompanied by a proof of possession of the public key of the client each time those are used.

In the same way, a secure communication when transferring bearers must be established in an effort to avoid an eavesdropper to obtain the token.

The relative length of the token compared to the request might be a problem in low-bandwith communications. No solution for this case is provided, although a stateful schema might be a solution for these cases.

Use of this schema is interesting for fleet management and OEMs that want to maintain the ecosystem whereas to keep some control over their vehicles or to access statistical data.

3.1.2 Connected Schema

The use of a connected schema implies that some of the actors in the ecosystem might interact between them in order to verify tokens or permit some of the requests.

In order for the system to work correctly, a secure and persistent connection must be established between the actors of the ecosystem that interact with each other. In case the verification of the tokens is delegated to the issuer of the token, it is possible to use a symmetric signature schema since the key will never leave the device that issues the token. If this verification is always delegated, the actor that issues the token could generate an opaque token or use a stateful strategy.

3.2 Depending on the Tokens Used

Tokens used during the protocol flow can follow the syntax described by VISSv2 protocol. This allows all actors in the protocol to know the content of the token and in case public signature is being used, the token can be verified. The tokens described in the VISSv2 protocol follows the JWT syntax, including some custom claims in order to be able to use them in the protocol. This makes these tokens readable, sacrificing efficiency since the token will be longer.

Proprietary tokens might be used, as described in some of the schemes described. These tokens will only be known by the issuer of the token, and most of the times, these tokens will be compared with cached or stored data in the issuer.

4 Appendix

4.1 Protocol Actors

- **Client** The client is the actor requesting to access the data in the Vehicle. This could be a cloud application storing data, the driver of the vehicle using the infotainment system of the vehicle or or the owner of the vehicle. In order to be able to describe the client, it is used its context. The context consists on three claims, the user of the application, the application, whether it is developed by the OEM or not and the device in which it is used depending on how it connects to the vehicle (remotely or directly).

- **Access Grant Token Server (AGTS).** The AGT Server a request including the client context (and a proof attesting that the client has the context it declares). It must verify the validity (and authenticity) of the context of the client and issue a signed Access Grant Token including that context.

- **Access Token Server (ATS).** The Access Token Server a request containing an AGT and a signal set or a purpose. The ATS must check the validity of the AGT received. After verifying the AGT, the ATS uses the context included in the AGT to check if the client can be authorized to access the purpose or set of signals requested. If the client can access the signals or purpose, an AT is generated including the authorization to access them. This is the behaviour described in VISSv2 specification and can be modified depending the access control schema required.

- **VISS Server.** Receives a request containing a set of signals requested by the client. This request must be accompanied with an authorization to access that data. This authorization, as described in this document, can be checked either by the VISS Server or delegated to other actor.

- **Policy Enforcement Point.** The policy enforcement point receives authorization requests that it must verify in order to give access or not to a signal. To perform this verification, it uses a set of policies that it must analyze.

4.2 Protocol Data

- **Purpose.** A purpose links a set of signals with a set of client contexts. This is used in order to generate tokens to access data. When a client requests a token containing a purpose, its context is compared with the contexts linked to the purpose. In the samse way, if a client requests data using a purpose, it is checked that the data can be accessed using that context.

- **Signal.** A signal represents data in the vehicle. It can be an actuator, a sensor or an attribute.
- **JSON Web Token (JWT).** A JWT is a token containing a set of claims signed. JSON Web Tokens are used in the protocol to include claims used to authenticate or give access to a set of data.
- **Access Grant Token (AGT).** The Access Grant Token is a JWT containing the client context and signed by the AGTS. The client uses AGTs to authenticate its context with the rest of the ecosystem.
- **Access Token (AT).** The Access Token is a JWT containing a purpose or a set of signals that the client can access. The client uses ATs to demonstrate it has authorization to access certain data.