


# Formal Analysis of C-ITS PKI protocols

Mounira Msahli<sup>1</sup><sup>a</sup>, Pascal Lafourcade<sup>2</sup><sup>b</sup>, and Dhekra Mahmoud<sup>2</sup>

<sup>2</sup>*Université Clermont Auvergne, CNRS, LIMOS, F-63000 Clermont, France*

<sup>1</sup>*Télécom Paris, LTCI, IP Paris, France*

*mounira.msahli@telecom-paris.fr, {pascal.lafourcade, dhekra.mahmoud}@uca.fr*


**Keywords:** C-ITS protocols; Formal Verification; Authentication; Privacy; Security; ProVerif


**Abstract:** Vehicular networking is gaining a lot of popularity and attraction from among the industry and academic research communities in the last decade. The communication between vehicles will lead to more efficient and secured roads because we will be able to provide information about traffic and road conditions to vehicle's drivers. However, ensuring the security of these networks and devices still remains a main major concern to guarantee the expected services. Secure Public Key Infrastructure (PKI) represents a common solution to achieve many security and privacy requirements. Unfortunately, current Cooperative Intelligent Transport Systems (C-ITS) PKI protocols were not verified in terms of security and privacy. In this paper, we propose a security analysis of C-ITS PKI protocols in the symbolic model using ProVerif. We formally modeled C-ITS PKI protocols based on the specifications given in the ETSI standard. We model C-ITS PKI protocols and formalize their security properties in the applied Pi-calculus. We used an automatic privacy verifier UKano to analyse Enrolment protocol. We found attacks on authentication properties, in Authorization and Validation protocols when considering a dishonest Authorization Authority (AA). We analysed proof results and we fixed identified attacks by introducing new parameters in protocol request.

## 1 Introduction

Connected vehicles are considered as a dynamic mobile communication system allowing gathering, sharing, processing, computing, and secure exchange of information between vehicles and infrastructure. These connected vehicles enable the evolution to next generation Intelligent Transportation Systems (ITSs) (Lamssaggad et al., 2021) (Severino et al., 2023). In (Contreras-Castillo et al., 2017), the authors give another definition of IoV; it is a platform that enables the exchange of information between the car and its surroundings through different communication media. Therefore, IoV is viewed as a wide area of vehicular network offering connectivity to a large number of online services and components. This concept extends the communication spaces of Infrastructure to Vehicles (I2V). It emphasizes the interaction among vehicles and its network infrastructure. Connected road users have to exchange data on their positions, status, speed, driving directions and the

required information to enable them to interact and to coordinate their behavior accordingly and safely. In complex traffic situations, for example, exchanged data facilitates vehicle navigation, reduces wait time and avoids delays. Many papers have discussed the security of C-ITS and have presented attacks using vulnerabilities detected, or likely to happen, in the C-ITS system. Examples of these attacks include the Sybil Attack, Location tracking attack, False message injection, Replay attack, Jamming attack and Traffic analysis attack (Lamssaggad et al., 2021) (Zhang et al., 2014) (Lu et al., 2019) (Szegedy et al., 2013). Accessing certain ITS services requires the use of PKI based authentication solution (Haidar et al., 2017). Main security standards did not address a complete detailed secure end-to-end mechanism to send requests to the PKI and receive the associated responses (Monteuuis et al., 2017). But, in literature, we could find several research works dealing with credential management for vehicular network. For example, the authors of (Simplicio et al., 2018) give an improved key management solution using butterfly key expansion process. Recently, the European Telecommunication Standards Institute (ETSI) has standardized technical specifications and

<sup>a</sup> <https://orcid.org/https://orcid.org/0000-0003-0331-493X>

<sup>b</sup> <https://orcid.org/https://orcid.org/0000-0002-4459-511X>

all requirements of communication protocol within the C-ITS PKI in the ETSI TS 102 941 standard (ETSI, ).

The C-ITS Trust Model (Lone et al., 2022) as defined in Certificate Policy for Deployment and Operation of European Cooperative Intelligent Transport Systems (C-ITS) (Platform, 2018) is based on a multiple root CAs architecture, where the root CA certificates are transmitted periodically to the Trust List Manager (TLM). The TLM issues the European Certificate Trust List that provides trust in the approved root CAs to all PKI participants. This Trust Model insures interoperability among several vehicular PKIs.

The main inconvenience drawback of this Trust Model is its centralization. The TLM shall retrieve RCAs from different PKIs to put it in one Certificate Trust List (CTL). Then to distribute the CTL periodically and regularly to all concerned relevant PKIs.

One home C-ITS PKI is composed of three authorities:

- Root Certification Authority (Root CA), is the highest level certification authority in the certification hierarchy. It is used to sign certificates for subordinate authorities (EA and AA).
- Enrolment Authority (EA), is responsible for the life cycle management of *Enrolment Certificates* (EC), which are long-term certificates used for the authentication of C-ITS stations to Authorisation Authority. EA has the duty to verify the canonical ID of the station sending a request.
- Authorization Authority (AA), is responsible for issuing and monitoring the use of *Authorization Tickets* (AT), which are short-lived anonymized certificates used by the corresponding C-ITS station to access permitted ITS services.

As mentioned in Figure 1, the ETSI TS 102 941 standard aims to specify a Public Key Infrastructure protocol for Intelligent Transport System. The protocol has made one of its requirement to provide privacy to stations (mostly vehicles). According to the standard, privacy is provided in two dimensions (relatively to other stations, but also relatively to authorities themselves).

Ultimately, the sectioning into three authorities is meant to provide privacy to the stations. The enrolment authority does not know the key the station is going to use to sign messages on the network, and therefore can't track its activity. The authorization authority shouldn't be able to link

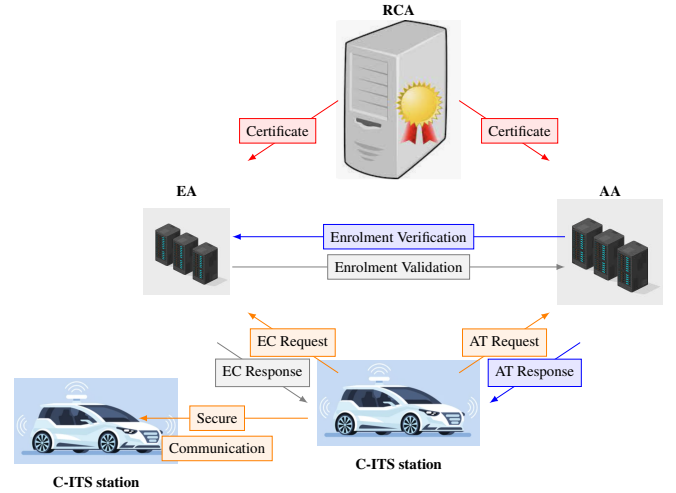


Figure 1: C-ITS PKI architecture

AT request between them, *i.e.*, it doesn't know if a request comes from a previously known station, or a new one. Therefore, an authorization authority trying to track the activity of a station should be lost as soon as the station renew its AT (or uses multiple AT simultaneously).

The different C-ITS PKI communications (requests/responses) are ruled by specific protocols, which are referred as C-ITS PKI protocols (Salin and Lundgren, 2023). This protocol is already implemented and used in European C-ITS system and needs a formal validation in order to avoid significant security and privacy concerns (Yoshizawa et al., 2023) (Chi et al., 2023). In this paper, we aim to verify the resistance of this protocol to passive and active (security and privacy) attacks using formal verification tools like AVISPA (Armando et al., 2005), ProVerif (Blanchet, 2016), Scyther (Basin et al., 2018) and Tamarin (Meier et al., 2013).

For the rest of this work, we choose the protocol verifier ProVerif (Barbosa et al., 2021) (Blanchet, 2014) and Ukano (Hirschi et al., 2019a), the automatic privacy verifier, as formal verification tools. In fact, ProVerif takes as input a protocol description in the applied Pi-Calculus, which is translated to a set of Horn clauses. ProVerif is fully automatic: the user only gives the specification of the protocol and the property to verify.

**Contribution.** In this paper, we present a new formalization for the C-ITS PKI protocols. We model C-ITS PKI protocol abstractly as three-party. When a C-ITS station requests an AT, the protocol involves two separate authorities. To the best of

our knowledge, we present the first formalization of several security properties for C-ITS PKI protocol. We categorize it into two main classes: (a) Authentication properties, including Request Origin Authentication, Request Authorship, and Response Authenticity, (b) Privacy properties and pseudonym indistinguishability. We develop our formal framework in the applied Pi-calculus wherein we propose a model for the C-ITS PKI's protocol and define all our properties. We validate our approach by analyzing: 1) Enrolment protocol, 2) Authorization protocol, and 3) Validation protocol. We model all exchanges in the applied Pi-calculus for verification using ProVerif. Our security analysis reveals several weaknesses in the second and the third protocols when dishonest parties collaborate with the attacker. We propose simple fixes to overcome those weaknesses.

The road-map of the paper is as follows: in Section 2, we give the related work. In Section 4, we model C-ITS PKI protocols and its security properties using the applied Pi-calculus. We verify symbolically the security and the privacy of three C-ITS PKI protocols and we discuss our results in Section 5 and 6. The conclusion is provided in Section 8.

## 2 Related Work

All C-ITS communications: Vehicle to Vehicle or Vehicle to Infrastructure or Vehicle to Everything could be a source of attack. In literature, several V2X protocols were verified using formal security tools. Bojjagani et al have designed a new IoV secure paradigm called AKAP IoV. The new scheme gives a mutual authentication, and key management among all VANET actors. The security level of AKAP-IoV was formally verified with Scyther and Tamarin tools(Bojjagani et al., 2022).Sutrala et al in(Sutrala et al., 2020) have designed a new conditional privacy preserving batch verification-based authentication in IoV. The security of proposed scheme were verified against a passive and active adversary through various security analysis.

Yashar and Vahid have proposed a security analysis of Secure Mutual Authentication Scheme and Key Exchange Protocol in Fog (SMAK-IoV). The authors used the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool to verify its security. The results show that SMAK-IoV is resistant to active and passive attacks.(Salami and Khajehvand, 2020).

Notation	Definition
$EA$	Enrolment Authority
$idEA$	Enrolment Authority's identifier
$pkEA$	$EA$ 's public key
$AA$	Authorization Authority
$idAA$	Authorization Authority's identifier
$pkAA$	$AA$ 's public key
$AT$	Authorization Ticket
$EC$	Enrolment Credential
$S$	ITS Station
$idS$	Station's identity
$skS$	Station's secret key
$pkS$	Station's public key
$pkS_i$	Public key of Station $S_i$
$sskS$	Station's private key used to sign
$spkS$	Public key corresponding to private key $sskS$
$k$	Symmetric key
$n$	Nonce
$pkV$	Station's public key appearing on AT
$skV$	Secret key corresponding to $pkV$
$As$	Authority receiving certificate request
$Aa$	Authority contacted by $As$
$id_{req}$	Identifier of a request
$id_{res}$	Identifier of a response
$CP$	CITS PKI protocol instance
$TLM$	Trust List Manager
$CTL$	Certificate Trust List
$CRL$	Certificate Revocation List

Table 1: Notation Table

(Yu et al., 2020) have demonstrated with formal validation tool that the proposed IoV-SMAP protocol can resist to security attacks and provides anonymity, and authentication .

In this paper, the main approach for protocol verification that we have adopted in our work is the symbolic model. Thanks to Needham and Shroeder (Needham and Schroeder, 1978) and Dolev and Yao (Dolev and Yao, 1983) (Lafourcade et al., 2009), cryptographic primitives are represented by function symbols and considered black boxes, messages are terms on these primitives, and the adversary is restricted to computing using only these primitives (Blanchet, 2014).

**Verifying protocols symbolically:** There exist numerous methodologies to formally verify protocols. Our work makes use of a particular modeling technique called "symbolic model" (Blanchet, 2014) that abstracts messages (including keys, nonces...) as terms and cryptographic primitives as black-box functions. Functions are related to each other through

equational theory, for example an encryption function  $enc$  and a decryption function  $dec$  could be related through the equation  $dec(enc(m, k), k) = m$  with  $k$  being a key and  $m$  a message. The attacker is able to intercept, modify and apply any functions to build terms using the equational theory, then send it to participants. In short, the network is the attacker. This model is commonly referred to as the Dolev-Yao model (Dolev and Yao, 1983). One of the tool developed to automate the study of protocols in the symbolic model is ProVerif (Blanchet, 2016) (Lafourcade and Puits, 2015). It takes as input a protocol modeled using a variant of pi-calculus, and is then able to search for attacks that break expected properties on it.

**Unlinkability:** One of the privacy feature that the protocol under study claims to offer is privacy through unlinkability (Baelde et al., 2020a) (Comon and Koutsos, 2017) (Baelde et al., 2020b) (Hirschi et al., 2019b). There exists numerous possible formal definitions for unlinkability. Here, we are going to use the definition proposed in (Hirschi and et al, 2016), since it fits quite well the expectations of our protocol. In our context, this definition means that an attacker should not be able to distinguish a situation where a station makes several requests to the PKI, and a situation where several stations make a single request. If this property is respected, we can easily conclude that an attacker cannot link messages coming from the same origin. In our case, we will use the main contribution introduced in (Hirschi and et al, 2016), a theorem that allows us to prove unlinkability by proving easier properties: if a protocol respects frame opacity and well-authentication.

**Frame opacity:** Roughly means that there is no relations visible to the attacker between messages; they are indistinguishable to randomness, and well-authentication means that any attempt by an attacker to tamper with the messages in order to leak information behind a conditional (say an authenticity test) should fail.

### 3 Threat Model

In this section, we consider the critical threats of our considered assumptions. Using the Dolev-Yao threat model, source and destination entities can exchange messages via exposed and insecure communication channel which is the internet. In our context, we consider the semi-honest malicious adversary threat model. Vehicles or C-ITS stations and PKI authorities are not considered as trusted entities. We con-

sider that all entities will follow the specified protocol in the real world as indicated in the ETSI standard[9], which implies that data structure of sent or received messages will not be changed. In our case, we consider that the adversary is able to eavesdrop, modify, replay, inject, or delete messages. The adversary could be a vehicle that replay requests or Authorisation authority that generate a response without the authorization validation. Here, the adversary can attempt "impersonation attacks" and "replay attacks".

## 4 Modeling C-ITS PKI protocols

### 4.1 Model

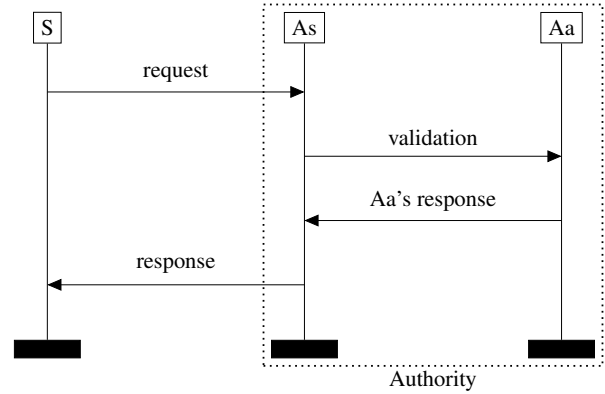


Figure 2: C-ITS PKI Protocols.

In this section, we model C-ITS PKI protocols in the applied Pi-calculus (Abadi et al., 2017). To perform the automatic protocol verification (Avalle et al., 2014), we use ProVerif (Blanchet, 2016) tool. Honest parties are modeled as processes in the applied Pi-calculus. They can exchange messages on public or private channels, create keys or fresh random values and perform tests and cryptographic operations, which are modeled as functions on terms with respect to an equational theory describing their properties.

The attacker has total control of the network (Dolev and Yao, 1983); messages may be read, modified, deleted or injected. Under the assumption of perfect cryptography, the attacker is only able to perform cryptographic operations when he is in possession of the right keys. C-ITS PKI protocol specifies the processes executed by a station  $S$  requesting a certificate to certification authorities  $As$  or  $Aa$ . We do not make assumptions about which authority issues the requested certificate. We only

specify which authority is being contacted by the station. We suppose that  $As$  is the authority contacted by the station and thus receiving the certificate request.  $Aa$  is the authority contacted by  $As$  for either a validation of the certificate request or for simply issuing it.

**Definition 4.1** (C-ITS PKI protocol). *A C-ITS PKI protocol is a tuple  $(S, As, Aa)$  where  $S$  is the process executed by the station;  $As$  is the process executed by the authority contacted by the station and  $Aa$  is the process executed by the authority contacted by  $As$ .*

**Definition 4.2** (C-ITS PKI instance). *In C-ITS PKI protocol, a C-ITS PKI instance is a closed process  $CP = \nu \tilde{n} \cdot (S\sigma_{idS}\sigma_{pkS}|As|Aa)$ , where  $\tilde{n}$  is the set of all restricted names;  $S\sigma_{idS}\sigma_{pkS}$  is the process run by the station, the substitutions  $\sigma_{idS}$  and  $\sigma_{pkS}$  specify the identity and the key to be certified respectively, and  $As$  (resp.  $Aa$ ) is the process run by the certification authority  $As$  (resp.  $Aa$ ).*

## 4.2 Security properties

The ETSI TS 102 941 standard introduces a major security requirement for C-ITS stations which is Privacy. The standard states that a C-ITS PKI protocol guarantees a C-ITS station's privacy when its pseudonym (or the short term certificate) could not be linked to its real identity. The architecture of the PKI actually separates the authorities involved in issuing pseudonym certificates (one for verifying ownership of an EC and the other for issuing certificates) to provide needed privacy for a C-ITS station. Neither EA nor AA should be able to link a pseudonym certificate to a user's real identity and neither EA nor AA should be able to link two (or more) pseudonym certificates to the same user.

To formalize the security properties required by C-ITS PKI protocols, we introduce a privacy property meant to prevent authorities or a third party from linking a station's pseudonym to its identity. We also introduce three authentication properties meant to ensure the association of the station's identity to its pseudonym.

**Authentication** In the following,  $idS$  is the station identity,  $pkS$  is the station's key to be certified,  $id_{req}$  is an identifier of the certificate request generated by the station and  $id_{res}$  is an identifier of the certificate response received by the station. Let  $set_{id} = \{idS, pkS\}$ ,  $p1 \in set_{id}$  and  $p2 \in set_{id} \setminus p1$ . We define the following events:

- $req(p1, p2, id_{req})$  is the event inserted into the  $S$

process at the location where a station  $S$  generates a certificate request  $id_{req}$ .

- $for(p1, id_{req})$  is the event inserted into the  $As$  process at the location where  $As$  receives, accepts, and generates the corresponding request for  $Aa$ .
- $val(p2, id_{req}, id_{res})$  is the event inserted into the  $Aa$  process where  $Aa$  receives and accepts the request coming from  $As$ .
- $res(p1, p2, id_{req}, id_{res})$  is the event inserted into the  $S$  process where a station receives and accepts a certificate response.

Events have the following structure "on every trace, the event  $e_2$  is preceded by the event  $e_1$ ". Here, the authentication concerns the source of the certificate request.

**Definition 4.3** (Request Origin Authentication). *The C-ITS PKI protocol ensures Request Origin Authentication if each occurrence of the event  $val(p2, id_{req}, id_{res})$  is preceded by an occurrence of the event  $for(p1, id_{req})$  and  $req(p1, p2, id_{req})$ .*

**Definition 4.4** (Request Authorship). *a C-ITS PKI protocol ensures Request Authorship if each event  $val(p2, id_{req}, id_{res})$  is preceded by a distinct occurrence of the event  $req(p1, p2, id_{req})$ .*

Our third property concerns the authenticity of a certificate response. Since  $As$  needs to contact  $Aa$  before responding to the station, this property ensures that whenever a station receives a certificate response,  $As$  has already contacted  $Aa$  and that  $Aa$  has responded.

**Definition 4.5** (Response Authenticity). *A C-ITS PKI protocol ensures Response Authenticity if for every occurrence of the event  $res(p1, p2, id_{req}, id_{res})$  there is an earlier distinct occurrence of the event  $val(p2, id_{req}, id_{res})$ .*

**Privacy** We model privacy property as observational equivalence (Blanchet, 2016). We use the labelled bisimilarity to express the equivalence between two processes. Informally, two processes are equivalent when an observer can not distinguish them apart. Let  $CP_S[-]$  be the process  $CP$  without  $S$ . Our privacy property concerns the Indistinguishability of the pseudonyms requested by the same station.

**Definition 4.6** (Pseudonym Indistinguishability). *A C-ITS PKI protocol ensures pseudonym indistinguishability if, for any C-ITS PKI process  $CP$ , any station with identity  $idS$  and any two pseudonyms  $pkS_1$  and  $pkS_2$ , we have:*

$$CP_S[S\sigma_{idS}\sigma_{pkS_1}] \approx_l CP_S[S\sigma_{idS}\sigma_{pkS_2}]$$

This property states that two processes with two different pseudonyms executed by the same station have to be observationally equivalent until the end of the protocol. This property requires that the authority knows  $idS$  to be honest. Otherwise, the property is trivially violated when the authority reveals the station's identity to the attacker.

### 4.3 Equational Theory

To capture the behaviour of the cryptographic primitives, we use the following equational theory:

$$sdec(senc(x, y, z), y, z) = x \quad (1)$$

$$adec(aenc(x, pk(y)), y) = x \quad (2)$$

$$checksign(sign(x, y), pk(y)) = x \quad (3)$$

$$checkhash(x, hash(x)) = true \quad (4)$$

$$checkhmac(hmac(x, y), x, y) = true \quad (5)$$

Equation (1) states that a message encrypted with a symmetric key and another encryption parameter can only be decrypted using those same parameters. This equation is used to abstractly model the *AES-CCM* encryption scheme in our case. The second equation 2 states that a message  $x$  encrypted with a public key can be decrypted using the corresponding secret key. Equation 3 models the verification of a signature using the corresponding public key. Equation (4) models the verification of a hashed value. Since we model a *hash* function as a one way function, one needs to know the message and the hashed value of the message to check the integrity of message. Equation (5) is about the verification of *hmac*.

## 5 Enrolment protocol: Security analysis

Enrolment protocol aims at authenticating and enrolling a station by issuing an EC to be used during the authorization phase. It is presented as a two-party protocol involving a C-ITS station requesting an EC and an enrolment authority responding by delivering or not the requested EC.

### 5.1 Description

**Enrolment request** To create an enrolment request, the C-ITS station begins by creating an InnerECRequest containing its canonical identifier  $idS$ , its verification public key  $spkS$  to be included in the certificate, the certificate format and the requested subject

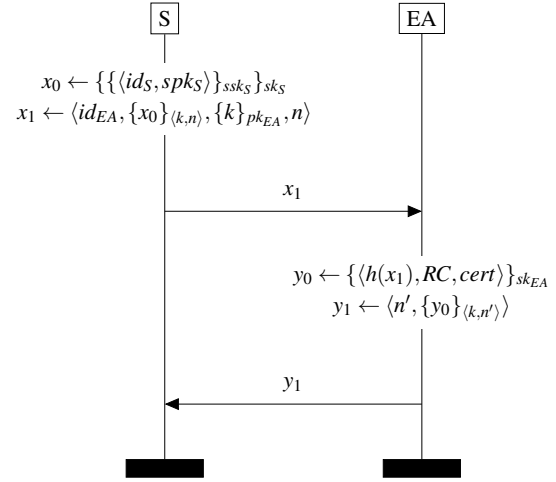


Figure 3: Enrolment Protocol

attributes. Then, a signed data structure is formed by signing the InnerECRequest with the private key  $sskS$  corresponding to the verification public key  $spkS$  for a proof of possession. Another signed data structure  $x_0$  is created by re-signing the indicated structure with the canonical private key  $skS$ . Finally, the C-ITS station encrypts its signed data with *AES-CCM* using a freshly generated symmetric key  $k$  and a nonce  $n$ . The station sends its encrypted and signed enrolment request along with the encrypted symmetric key (encrypted with EA's public key), the used nonce and EA's public certificate identifier  $idEA_v$  (see Fig.3).

**Enrolment Response** When EA receives an enrolment request, it begins by deciphering the encrypted symmetric key  $k$ . Then, it decrypts the encrypted data structure, checks the signatures performed by the C-ITS station after verifying its identity and retrieves the verification public key  $spkS$ . If the verification succeeds, EA creates an Enrolment Credential  $cert$  and sets to 0 its response code  $RC$ , which corresponds to a positive response. EA creates an InnerECResponse containing the request hash, a response code and a certificate (in case of a positive response). Then, a signed data is generated by signing the InnerECResponse with the private key  $skEA$ . The data is finally encrypted with *AES-CCM* using the same symmetric key  $k$  used to encrypt the request and a freshly generated nonce  $n'$  (see Fig. 3).

### 5.2 Model in the Applied Pi-Calculus

We expressed the behaviour of the participants using the processes depicted in Figures 4 and 5.

$EP \doteq$

- 1: (\* private keys and identifiers \*)
- 2:  $\nu idS \cdot \nu skS \cdot \nu idEA \cdot \nu skEA \cdot$
- 3: (\* public keys \*)
- 4: let  $pkEA = pk(skEA)$  in let  $pkS = pk(skS)$  in
- 5: (\* public key disclosure \*)
- 6:  $out(c, pkEA) \cdot$
- 7: (\* Station process — Enrolment Authority process \*)
- 8:  $(\nu sskS \cdot P_S \mid !P_{EA})$

Figure 4: The main process

$P_S \doteq$

- 1: let  $InnerECRequest = (idS, pk(sskS))$  in
- 2: let  $x_0 = sign(sign(InnerECRequest, sskS), skS)$  in
- 3:  $\nu k \cdot \nu n \cdot$
- 4: let  $x_1 = senc(x_0, k, n)$  in
- 5: let  $x_2 = aenc(k, pkEA)$  in
- 6:  $out(c, (idEA, x_1, x_2, n)) \cdot$
- 7:  $in(c, (x_3, x_4))$
- 8: let  $(x_h, x_r, x_c) = checksign(sdec(x_3, k, x_4), pkEA)$  in
- 9: if  $checkhash((idEA, x_1, x_2, n), x_h) = true$  then
- 10: if  $x_r = true$  then
- 11: if  $checksign(x_c, pkEA) = pk(sskS)$  then 0

$P_{EA} \doteq$

- 1:  $in(c, (y_0, y_1, y_2, y_3)) \cdot$
- 2: if  $y_0 = idEA$  then
- 3: let  $k = aenc(y_2, skEA)$  in
- 4: let  $y_4 = sdec(y_1, k, y_3)$  in
- 5: let  $(idS, spkS) = getmess(getmess(y_4))$  in
- 6: if  $checksign(checksign(y_4, pkS), spkS) = (idS, spkS)$  then
- 7: let  $InnerECResponse = (hash((idEA, y_1, y_2, y_3)), true, sign(spkS, skEA))$  in
- 8:  $\nu n \cdot$
- 9:  $out(c, (senc(sign(InnerECResponse, skEA), k, n), n))$

Figure 5: Station and Enrolment Authority processes.

### 5.3 Analysis

In addition to ProVerif, we have used the automatic privacy verifier UKano (Hirschi et al., 2019a) which checks the *unlinkability* and the *anonymity* of two-party protocols by verifying two properties namely *Frame-Opacity* and *Well-Authentication*. Informally, *Frame-Opacity* requires that in any execution of the protocol, the attacker should not be able to distinguish the outputs from pure randomness. *Well-Authentication* prevents the attacker from

obtaining valid information through the outcome of any conditionals. Hirschi *et al.* demonstrated that *Frame-Opacity* and *Well-Authentication* imply *unlinkability* and *anonymity* (Hirschi et al., 2019a).

We present the results of our analysis of Enrolment protocol’s security properties in Table 2.

Property	Result	Time
Request Authorship	✓	< 1 s
Response Authenticity	✓	< 1 s
Pseudonym Indistinguishability	✓	< 1 s
Well-Authentication	✓	< 1 s
Frame Opacity	×	< 1 s

Table 2: Results of authentication and privacy properties’ analysis.

**Authentication** Since Enrolment protocol involves only a single authority, we do not need to check Request Origin Authentication property. Instead, we only check Request Authorship and Response Authenticity properties. In fact, when Request Authorship holds, Request Origin Authentication is trivially verified.

**Privacy** Since Enrolment Authority knows both the identity of the station and the certified key, we do not need to check pseudonym indistinguishability with regard to Enrolment Authority. We only check this property with regard to an external attacker and we found that it is verified. Moreover, we used the automatic privacy verifier UKano which checks automatically *unlinkability* and *anonymity* of both the station and Enrolment Authority by checking *Frame-Opacity* and *Well-Authentication*. We found that Enrolment protocol fails when checking *Frame Opacity*. In fact, the identity of EA is sent in plain text in the enrolment request and since UKano checks the *anonymity* and the *unlinkability* of both the sender and the receiver, it is obvious that Enrolment protocol does not guarantee the privacy of EA. We do not consider this as an attack because the standard does not claim to preserve *anonymity* nor *unlinkability* of EA. However, since every C-ITS station is associated to a single EA, an attacker would collect data about stations belonging to the same manufacturer and this may reveal some information about the stations’ identity or private information.

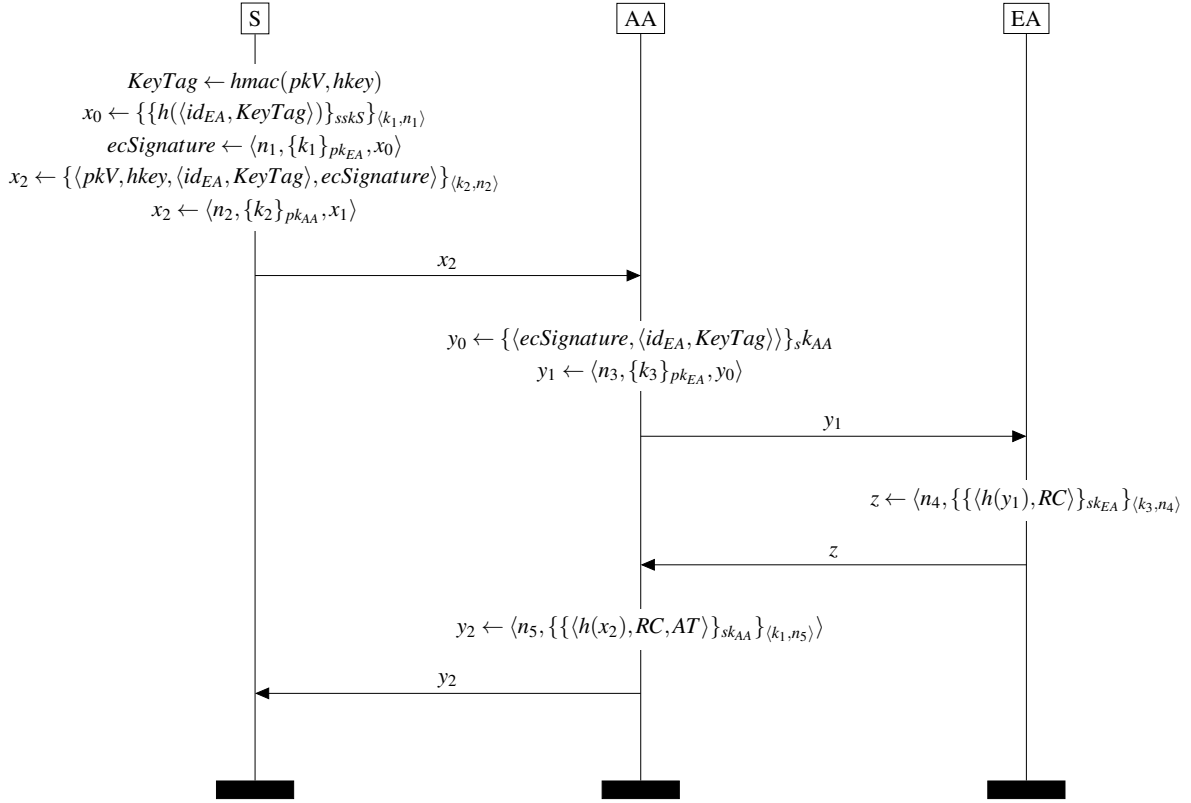


Figure 6: Authorization Protocol

## 6 Authorization-Validation protocol: Security analysis

The Authorization-Validation protocol is presented in two steps in the ETSI standard (see figure 6). Firstly, the C-ITS station requests an AT from the authority AA, which could respond positively if the station is eligible to get the requested certificate or with a negative response if not. The validation protocol involves two authorities AA and EA. AA requests the validation of EA to deliver or not the requested certificate by the station. EA responds positively if the authority is able to authenticate the station and if the station is permitted to get access to the requested services. Otherwise, it responds negatively.

We have modeled this protocols as one single three-party protocol, i.e., the response of AA in the Authorization protocol depends on the response of EA in the Validation protocol.

### 6.1 Description

Authorization-Validation protocol aims at authorizing an enrolled station to use services privately over the

$AVP \doteq$

- 1: (\* private keys \*)
- 2:  $\forall skS \cdot \forall skAA \cdot \forall skEA \cdot$
- 3: (\* public keys \*)
- 4: let  $spkS = pk(skS)$  in let  $pkAA =$
- 5:  $pk(skAA)$  in let  $pkEA = pk(skEA)$  in
- 6: (\* public keys disclosure \*)
- 7:  $out(c, pkAA) \cdot out(c, pkEA) \cdot$
- 8: (\* Station — Authorization Authority — Enrolment Authority \*)
- 9:  $(!vskV \cdot P_S \mid !P_{AA} \mid !P_{EA})$

Figure 7: The main process.

Property	Result	Time
Request Origin Authentication	✓	< 1 s
Request Authorship	×	< 1 s
Response Authenticity	✓	< 1 s
Request Origin Authentication*	×	< 1 s
Request Authorship*	×	< 1 s
Response Authenticity*	×	< 1 s
Pseudonym Indistinguishability	✓	< 1 s

Table 3: Results of security properties' analysis. *Property\** is the property checked with dishonest Authorization Authority.



```

 $P_S \doteq$ 
1: let  $pkV = pk(skV)$  in
2:  $vhkey$ .
3: let  $KeyTag = hmac(pkV, hkey)$  in
4: let  $x_0 = (idEA, KeyTag)$  in
5: let  $x_1 = sign(hash(x_0), sskS)$  in
6:  $vk_{EA}.vn_{EA}$ .
7: let  $ecSignature = (senc(x_1, k_{EA}, n_{EA}), aenc(k_{EA}, pk_{EA}, n_{EA}))$  in
8: let  $ATRequest = (pkV, hkey, x_0, ecSignature)$  in
9:  $vk_{AA}.vn_{AA}$ .
10: let  $k_c = aenc(k_{AA}, pk_{AA})$  in
11: let  $AuthorizationReq = (idAA, senc(ATRequest, k_{AA}, n_{AA}), k_c, n_{AA})$  in
12:  $out(c, AuthorizationReq)$ .
13:  $in(c, (x_2, x_3))$ 
14: let  $(hr, res, AT) = checksign(sdec(x_2, k_{AA}, x_3), pk_{AA})$  in
15: if  $checkhash(AuthorizationReq, hr) = true$  then
16: if  $res = true$  then
17: if  $checksign(AT, pk_{AA}) = pkV$  then 0

```

Figure 8: Station's process.

network to obtain an AT. We describe it as a three-party protocol involving a C-ITS station, an Authorization Authority and an Enrolment Authority. In short, the C-ITS station sends an AT request to AA which checks the validity of the request and then sends a validation request to EA. EA authenticates the station and sends its response to AA. The AA responds to the station accordingly.

**Authorization request** To create an authorization request, the C-ITS station should first generate a key pair ( $skV, pkV$ ) such that  $pkV$  is to be provided to the AA included in the AT. It should also generate a random key  $hmac$  – key to calculate a tag on  $pkV$  using an  $hmac$  function. We refer to this tag as  $KeyTag$ . Next, the station creates a  $SharedATRequest$  structure containing  $idEA$  identifying the EA to be contacted for validation, the  $KeyTag$ , the certificate format, and the requested subject attributes. Then, the station signs the hash of the  $SharedATRequest$  with its private enrolled key  $sskS$  and creates an encrypted data structure using  $AES - CCM$  and the EA public encryption key  $pk_{EA}$ . Let  $ecSignature$  be the signed and encrypted  $SharedATRequest$ . The station builds an  $InnerATRequest$  structure containing  $pkV$ ,  $hmac$  key, the  $SharedATRequest$ , and the  $ecSignature$ . The  $InnerATRequest$  is later encrypted and sent to the AA along with the encryption parameters.

**Authorization-Validation request** AA decrypts the authorization request and obtains the  $InnerATRequest$ . It checks the value of  $KeyTag$ . If the tag is not valid, it responds negatively. If not, it forwards the  $ecSignature$  and  $SharedATRequest$  to the corresponding EA after signing and encrypting its message.

**Authorization-Validation response** EA decrypts the Authorization-Validation request and checks the signature of AA. Then, it decrypts the  $ecSignature$  and checks the hash of the  $SharedATRequest$ . Finally, it checks the signature of the station over the hash of  $SharedATRequest$  and verifies whether it is enrolled or not and if it has permission to access the desired subject attributes. If the verification succeeds, EA creates an Authorization Validation Response structure containing the request hash, a positive response code ( $= 0$ ), and the confirmed subject attributes. This structure is then signed, encrypted, and sent to AA.

**Authorization response** Based on the  $KeyTag$  verification and EA's response, AA creates an  $AuthorizationResponse$  structure containing the request hash, a response code and an AT. It signs it and encrypts it using the symmetric key sent by the station and a freshly generated nonce.

## 6.2 Model in the applied Pi-Calculus

We expressed the behaviour of our participants using the processes depicted in Figures 7, 8, 9 and 10.

**Authentication** First, we analyze the authentication properties using honest parties in ProVerif and find that Request Origin Authentication and Response Authorship are successfully verified. However, Request Authorship property is not verified because, theoretically, an external attacker could replay a validation request sent by the Authorization Authority until the request is no longer accepted by the Enrolment Authority. We do not consider this as an attack because the ETSI TS 102 731 (ETSI, 2010) standard suggested a countermeasure by providing Replay Protection services based on a timestamp. We have also analyzed the same properties considering a dishonest Authorization Authority and found attacks on all authentication properties.

**Attack Against Request Origin Authentication** We suppose that an enrolled C-ITS station has requested an AT with  $pkV$ ,  $spkS$  and  $ecSignature$  parameters from a dishonest Authorization Authority

$P_{AA} \doteq$   
 1:  $in(c, (y_0, y_1, y_2, y_3))$   
 2: if  $y_0 = id_{AA}$  then  
 3: let  $k_S = adec(y_2, sk_{AA})$  in  
 4: let  $(pkV, hmac - key, SharedATRequest, ecSignature) = sdec(y_1, k_S, y_3)$  in  
 5: let  $(id_{EA}, KeyTag) = SharedATRequest$  in  
 6: if  $checkhmac(KeyTag, pkV, hmac - key) = true$  then  
 7:  $\forall k \cdot \forall n \cdot$   
 8: let  $k_c = aenc(k, pk_{EA})$  in  
 9: let  $ValidationReq = (id_{EA}, senc(sign((KeyTag, ecSignature), sk_{AA}), k, n), k_c, n)$  in  
 10:  $out(c, ValidationReq)$   
 11:  $in(c, (y_4, y_5))$   
 12: if  $checksign(sdec(y_4, k, y_5), pk_{EA}) = (hash(ValidationReq), true)$  then  
 13: let  $AT = sign(pkV, sk_{AA})$  in  
 14: let  $AuthorizationResponse = (hash((y_0, y_1, y_2, y_3)), true, AT)$  in  $\forall n_S \cdot$   
 15:  $out(c, (senc(sign(AuthorizationResponse, pk_{AA}), k_S, n_S), n_S))$

Figure 9: Authorization Authority's process.

$P_{EA} \doteq$   
 1:  $in(c, (z_0, z_1, z_2, z_3))$   
 2: if  $z_0 = id_{EA}$  then  
 3: let  $k = adec(z_2, sk_{EA})$  in  
 4: let  $requestSigned = sdec(z_1, k, z_3)$  in  
 5: let  $(KeyTag, ecSignature) = checksign(requestSigned, pk_{AA})$  in  
 6: let  $(z_4, z_5, z_6) = ecSignature$  in  
 7: if  $checkhash(KeyTag, checksign(sdec(z_4, adec(z_5, sk_{EA}), z_6), spk_S)) = true$  then  $\forall n \cdot$   
 8:  $out(c, (senc(sign((hash((z_0, z_1, z_2, z_3)), true), sk_{EA}), k, n), n))$

Figure 10: Enrolment Authority's process.

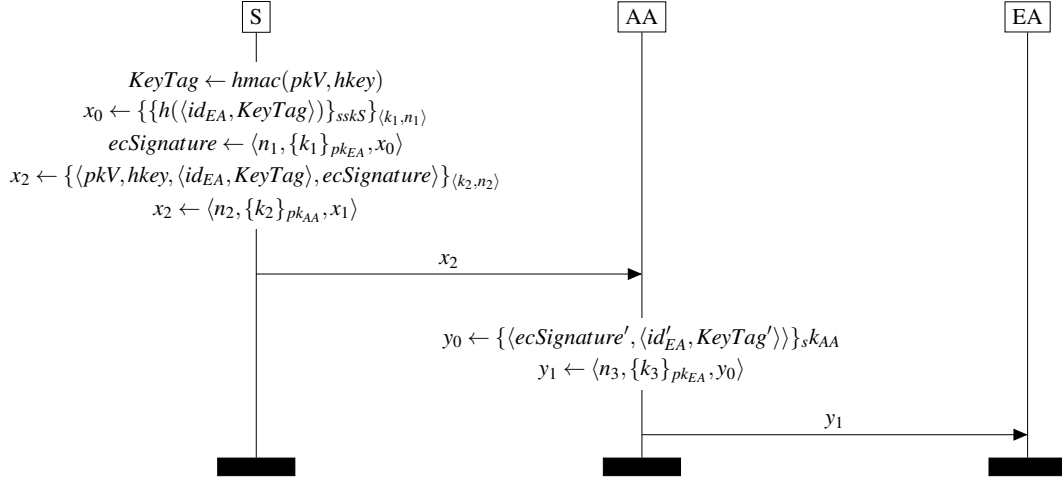


Figure 11: Attack against request origin authentication.

which did not forward its request to the corresponding Enrolment Authority but instead kept those parameters in memory. A dishonest Authorization Authority could easily modify requests by injecting saved data from previous requests. AA could use the identity of any C-ITS station to create Authorization Tickets without its consent and without even its

knowledge. For example, in case of misbehaving the pseudonym of the vehicle could be linked to a false C-ITS station identity. We recall that structures in the C-ITS standard are set up with a generation time. Therefore, this attack remains valid until the expiry time of saved structures. The sequence diagram of this attack is depicted in Figure 11.

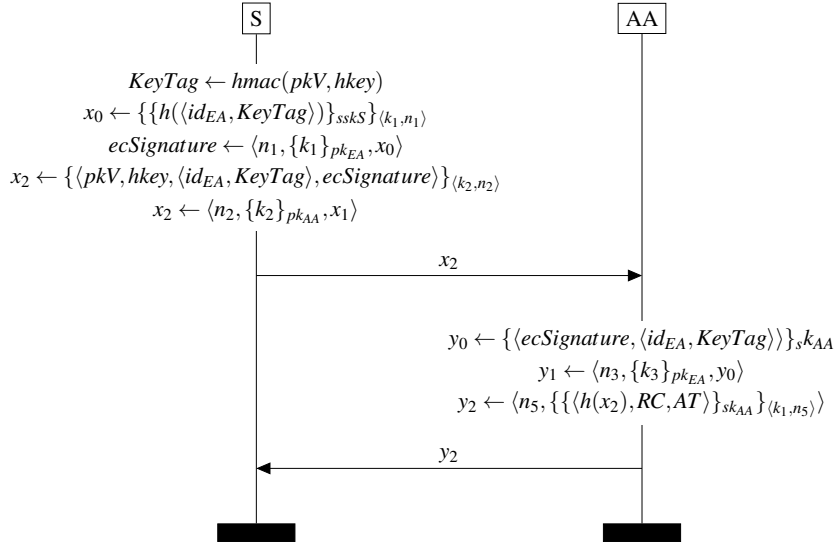


Figure 12: Attack against response authenticity.

**Attack Against Response Authenticity** First, Authorization-Validation protocol does not provide any means for a C-ITS station to verify authorization response authenticity. Authorization responses come without any proof on whether or not they have been approved by Enrolment Authority. Authorization Authority may respond negatively to a valid AT request whereas Enrolment Authority (EA) has already approved it. Secondly, It is undeniable that Authorization Authority could deliver Authorization Tickets to any C-ITS station. In the description of the Authorization-Validation protocol, Authorization Authority has total control over Authorization Tickets issuing. Therefore, it may respond positively to an AT request without forwarding a validation request to AA. This means that Authorization Authority may deliver an AT to a non-enrolled or a revoked C-ITS station. The sequence diagram of this attack is depicted in Figure 12. To fix this attack, we propose to include, in the certificate response, a response from EA intended for the station. Since the station has already shared a fresh symmetric key with EA, we use this key to encrypt EA's response which includes the *KeyTag* along with a response code.

**Attack Against Request Authorship** Violating this property seems void of any real threat when dealing with honest participants; an external attacker would only replay the validation request sent by AA and get an encrypted response from EA. Moreover, as previously mentioned, the Replay Protection Ser-

vices may reject a replayed request. But in the case of a dishonest AA, the threat becomes very serious as it could generate validation requests based on previous received requests and would encrypt them with freshly generated parameters so that they would not be detectable by the Replay Protection Services. For example, AA may deliver an AT for its own use by replaying a previous validation-authorization request with an enrolled C-ITS station parameter. This scenario may seem unrealistic because a dishonest AA could easily generate an AT for its own use. But when AA obtains a validation from EA, means that the protocol has been "theoretically" executed honestly. Such misbehavior has fewer chances of being disclosed or reported.

## 7 Recommendations

To fix the attack against response authenticity, we propose to include, in the certificate response, a response from EA intended for the station. Since the station has already shared a fresh symmetric key with EA, we use this key to encrypt EA's response which includes the *KeyTag* along with a response code.

To repair the attack against request authorship, we propose that the EA compare received *KeyTag* to a list of recently received *KeyTags*. The *KeyTag* is used to verify the freshness and to establish the direct link between the vehicle and the EA. If a same one has been used recently, EA rejects the validation.

## 8 Conclusion

Protocols in vehicular network are susceptible to various and several security threats and attacks. In fact, sensitive information can be transmitted using a risky and insecure channel. In this paper, we formally analysed C-ITS PKI protocols in the symbolic model based on the applied Pi-calculus using Pro-Verif and Ukano verification tools. Therefore, we used the automated verification tool ProVerif. We formally modeled C-ITS PKI protocols based on the specifications given in the ETSI standard. We used an automatic privacy verifier UKano to analyse all PKI protocols. We found attacks on authentication properties, in Authorization and Validation protocols when considering a dishonest Authorization Authority (AA). We fixed two attacks by including a response (*KeyTag*) from Enrolment Authority EA to the C-ITS station in the certificate response. Via this work, we demonstrated that the actual European security and interoperability and ETSI PKI architecture and protocols for C-ITS used in different European countries suffer from several security problems that could be verified and resolved. As a future work, more simulation can be performed using the SUMO tool or NS3 to verify proposed solutions. We could extend formal verification to other V2X protocols.

## REFERENCES

- Abadi, M., Blanchet, B., and Fournet, C. (2017). The applied pi calculus: Mobile values, new names, and secure communication. *Journal of the ACM (JACM)*, 65(1):1–41.
- Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J., Drielsma, P. H., Héam, P.-C., Kouchnarenko, O., Mantovani, J., et al. (2005). The avispa tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification*, pages 281–285. Springer.
- Avalle, M., Pironi, A., and Sisto, R. (2014). Formal verification of security protocol implementations: a survey. *Formal Aspects of Computing*, 26(1):99–123.
- Baelde, D., Delaune, S., and Moreau, S. (2020a). A method for proving unlinkability of stateful protocols. In *2020 IEEE 33rd Computer Security Foundations Symposium (CSF)*, pages 169–183. IEEE.
- Baelde, D., Delaune, S., and Moreau, S. (2020b). *A Method for Proving Unlinkability of Stateful Protocols*. PhD thesis, Irista.
- Barbosa, M., Barthe, G., Bhargavan, K., Blanchet, B., Cremers, C., Liao, K., and Parno, B. (2021). Sok: Computer-aided cryptography. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 777–795. IEEE.
- Basin, D., Cremers, C., and Meadows, C. (2018). Model checking security protocols. In *Handbook of Model Checking*, pages 727–762. Springer.
- Blanchet, B. (2014). Automatic verification of security protocols in the symbolic model: The verifier proverif. *Foundations of Security Analysis and Design VII*, 8604(3):54–87.
- Blanchet, B. (2016). Modeling and verifying security protocols with the applied pi calculus and proverif. *Foundations and Trends® in Privacy and Security*, 1(1-2):1–135.
- Bojjagani, S., Reddy, Y. P., Anuradha, T., Rao, P. V., Reddy, B. R., and Khan, M. K. (2022). Secure authentication and key management protocol for deployment of internet of vehicles (ioV) concerning intelligent transport systems. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24698–24713.
- Chi, L., Msahli, M., Memmi, G., and Qiu, H. (2023). Public-attention-based adversarial attack on traffic sign recognition. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pages 740–745. IEEE.
- Comon, H. and Koutsos, A. (2017). Formal computational unlinkability proofs of rfid protocols. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 100–114. IEEE.
- Contreras-Castillo, J., Zeadally, S., and Guerrero-Ibañez, J. A. (2017). Internet of vehicles: architecture, protocols, and security. *IEEE internet of things Journal*, 5(5):3701–3709.
- Dolev, D. and Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208.
- ETSI, T. Etsi ts 102 941 v1. 1.1-intelligent transport systems (its); security; trust and privacy management,” standard, tc its, 2012.
- ETSI, T. (2010). Etsi ts 102 731 v1. 1.1-intelligent transport systems (its); security; security services and architecture. *Standard, TC ITS*.
- Haidar, F., Kaiser, A., and Lonc, B. (2017). On the performance evaluation of vehicular pki protocol for v2x communications security. In *2017 IEEE 86th vehicular technology conference (VTC-Fall)*, pages 1–5. IEEE.
- Hirschi, L., Baelde, D., and Delaune, S. (2019a). A method for unbounded verification of privacy-type properties. *Journal of Computer Security*, 27(3):277–342.
- Hirschi, L., Baelde, D., and Delaune, S. (2019b). A method for unbounded verification of privacy-type properties. *Journal of Computer Security*, 27(3):277–342.
- Hirschi, L. and et al, B. (2016). A method for verifying privacy-type properties: the unbounded case. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 564–581. IEEE.
- Lafourcade, P. and Puys, M. (2015). Performance evaluations of cryptographic protocols verification tools dealing with algebraic properties. In *International*

- Symposium on Foundations and Practice of Security*, pages 137–155. Springer.
- Lafourcade, P., Terrade, V., and Vigier, S. (2009). Comparison of cryptographic verification tools dealing with algebraic properties. In *International Workshop on Formal Aspects in Security and Trust*, pages 173–185. Springer.
- Lamssaggad, A., Benamar, N., Hafid, A. S., and Msahli, M. (2021). A survey on the current security landscape of intelligent transportation systems. *IEEE Access*, 9:9180–9208.
- Lone, F. R., Verma, H. K., and Sharma, K. P. (2022). Recommender credibility-based trust model for secure v2x communication. In *2022 5th International Conference on Computational Intelligence and Networks (CINE)*, pages 1–6. IEEE.
- Lu, Z., Qu, G., and Liu, Z. (2019). A survey on recent advances in vehicular network security, trust, and privacy. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):760–776.
- Meier, S., Schmidt, B., Cremers, C., and Basin, D. (2013). The tamarin prover for the symbolic analysis of security protocols. In *International conference on computer aided verification*, pages 696–701. Springer.
- Monteuuis, J. P., Hammi, B., Salles, E., Labiod, H., Blancher, R., Abalea, E., and Lonc, B. (2017). Securing pki requests for c-its systems. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8. IEEE.
- Needham, R. M. and Schroeder, M. D. (1978). Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999.
- Platform, C.-I. (2018). Certificate policy for deployment and operation of european cooperative intelligent transport systems (c-its). *Journal of Computer Security*, RELEASE 1.1.
- Salami, Y. and Khajehvand, V. (2020). Smak-iov: secure mutual authentication scheme and key exchange protocol in fog based iov. *Journal of Computer & Robotics*, 13(1):11–20.
- Salin, H. and Lundgren, M. (2023). A gap analysis of the adoption maturity of certificateless cryptography in cooperative intelligent transportation systems. *Journal of Cybersecurity and Privacy*, 3(3):591–609.
- Severino, R., Simão, J., Datia, N., and Serrador, A. (2023). Protecting hybrid its networks: A comprehensive security approach. *Future Internet*, 15(12):388.
- Simplicio, M. A., Cominetti, E. L., Patil, H. K., Ricardini, J. E., and Silva, M. V. M. (2018). The unified butterfly effect: Efficient security credential management system for vehicular communications. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–8. IEEE.
- Sutrala, A. K., Bagga, P., Das, A. K., Kumar, N., Rodrigues, J. J., and Lorenz, P. (2020). On the design of conditional privacy preserving batch verification-based authentication scheme for internet of vehicles deployment. *IEEE Transactions on Vehicular Technology*, 69(5):5535–5548.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
- Yoshizawa, T., Singelée, D., Muehlberg, J. T., Delbruel, S., Taherkordi, A., Hughes, D., and Preneel, B. (2023). A survey of security and privacy issues in v2x communication systems. *ACM Computing Surveys*, 55(9):1–36.
- Yu, S., Lee, J., Park, K., Das, A. K., and Park, Y. (2020). Iov-smap: Secure and efficient message authentication protocol for iov in smart city environment. *IEEE access*, 8:167875–167886.
- Zhang, K., Liang, X., Lu, R., and Shen, X. (2014). Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal*, 1(5):372–383.