A formally verified Trust and Reputation Model for VANET

Giuseppe Primiero

***Abstract*—Vehicle Ad Hoc Networks (VANET) are becoming an important part of intelligent transportation systems. In this context, security requirements need to rely on a combination of agents' reputation and trust relations over the messaging infrastructure in order to maintain a dynamic and safe behaviour evaluation. Ontologically grounded approaches seem to offer optimal coverage and semantic foundation but formal correctness of the model and the safety of transitive operations within the infrastructure remain unaddressed, with potentially disastrous effects. In this paper we offer a proof-theoretical interpretation of such a reputation and trust model for VANET which allows for a formal verification through translation in the Coq proof assistant and guarantees security of transitive information transmissions.**

1. Introduction

This paper provides a proof-theoretical translation of the trust and reputation model for VANET offered in [**?**] in an extension of the natural deduction calculus `SecureND` from [**?**] and [**?**]. The aim is, first of all, to show that the trust properties instantiated though `SecureND` faithfully reflect those in transmission across the VANET network; accordingly, non-trustworthy interactions can be proven to be such through a proof-checking method. On a higher level, the model offered by `SecureND` has been proven formally correct through its translation to a Coq library, and as such the present translation guarantees a similar property for the whole VANET model.

2. Related Work

3. (un)`SecureND`

(un)`SecureND` is a natural deduction calculus defining trust, mistrust and distrust protocols introduced in [**?**] for the positive fragment and in [**?**] for the negation complete extension. We offer here a slightly modified version adapted for the VANET network. In particular, the present version introduces: contexts as sets of sets; formulas with multiple indices to account for service and message numbers; ranking on service characteristics. We start with introducing the language of our logic:

Definition 1 (Syntax of (un)`SecureND`)**.**

$$
\begin{array}{l}
\mathcal{A}^{\prec} := \{\mathcal{V}, \mathcal{R}\}\\
\mathcal{V} := \{v_1 \prec \ldots \prec v_n\}\\
\mathcal{R} := \{rsu_1 \ldots rsu_n\}\\
\mathcal{S} := \{S_1 \leq \cdots \leq S_n\}\\
\mathcal{C} := \{C_1 \leq \cdots \leq C_n\}\\
\mathcal{M}^{\mathcal{A}} := a^{\mathcal{A}}_{S_i,C_j} \mid \neg\phi^{\mathcal{A}}_{i,j} \mid \phi^{\mathcal{A}}_{i,j} \rightarrow \phi^{\mathcal{A}}_{k,l} \mid \phi^{\mathcal{A}}_{i,j} \wedge \phi^{\mathcal{A}}_{k,l} \mid \phi^{\mathcal{A}}_{i,j} \vee \phi^{\mathcal{A}}_{k,l} \mid \bot\\
mode := Read(\mathcal{M}^{\mathcal{A}}) \mid Write(\mathcal{M}^{\mathcal{A}}) \mid Trust(\mathcal{M}^{\mathcal{A}})\\
RES := \mathcal{M}^{\mathcal{A}} \mid mode \mid \neg RES\\
\Gamma^{\mathcal{A}} := \phi^{\mathcal{A}}_{i,j} \mid \phi^{\mathcal{A}}_{i,j} < \phi^{\mathcal{A}}_{k,l} \mid \Gamma^{\mathcal{A}}; \phi^{\mathcal{A}}_{i,j}
\end{array}
$$

3.1. Services, Messaging and Protocols

$\mathcal{M}^{\mathcal{A}}$ is a set of boolean formulae, closed under connectives, expressing messages. The language includes $\bot$ to express conflicts. $\mathcal{A}$ is the set of agents issuing messages and including vehicles $\mathcal{V}$ and RSUs $\mathcal{R}$. Messages are then signed by agents (vehicles or RSU) generating them and with a service number from $\mathcal{S}$ and a message number, so that: $\phi^{v_i}_{i,j}$ says that message ϕ about service S_i and characteristic j is generated by vehicle v_i. We assume here and throughout that both services $\mathcal{S}$ and characteristics $\mathcal{C}$ of services are given as posets. To simplify notation, a message ${}^{v_i}_{S_i,C_j}$ is usually abbreviated as ${}^{v_i}_{i,j}$. $mode$ is a variable for reading, writing and trusting messages, closed under negation. An agent profile $\Gamma^{\mathcal{A}}$ is the current list of all messages collected by the agent either from other agents or from various available sensors and other networks. For the present purposes, the latter ones will be indexed at their first vehicle or RSU collecting it.

Definition 2 (Formulas)**.** *A formula $\Gamma^v_i \vdash_{\mathtt{s}} \phi^{v_j}_{i,k}$ says that a message ϕ with number k signed from agent v_j about service i is validly accessed at step $\mathtt{s} \geq 0$ under the profile of agent v_i.*

Definition 3 (Validity)**.** *A formula $\vdash_{\mathtt{s}} \phi^{v_j}_{i,k}$ says that a message ϕ with number k signed from agent v_j about service i holds for any agent's profile at step $\mathtt{s}$.*

Messages satisfies a ranking for each agent based on that of services and characteristics:

Definition 4. *$\phi^{v_j}_{i,k} < \phi^{v_j}_{i,l}$ iff $C_k \leq C_k \in S_i$*

The order relation between service characteristics induces therefore validity under profile: if a characteristic i is essential to another one l with respect to a service i for an agent v_j, then that agent will be required to obtain a value for i in order to validly access a value for l. This order relation between service characteristics can then be lifted at the level of services. A partial order relation $\leq$ over $\mathcal{S} \times \mathcal{S}$ intuitively expresses that an order is satisfied across services:

Definition 5. $S_i \leq S_j$ *iff* $\exists \phi^{v_j}_{i,k}, \phi^{v_j}_{j,l}$ *s.t.* $\phi^{v_j}_{i,k} < \phi^{v_j}_{j,l}$.

$S_i < S_j$ means therefore that a service S_i is essential for a service S_j if there is a message concerning a characteristic of the latter which requires a message about a characteristic of the former. The partial order allows for branching in the hierarchy, so that e.g. $S_i < S_j < S_k$ and $S_i < S_j < S_l$, i.e. packages in S_k, S_l both require service S_j and transitively S_i, but they have no requirement from each other.

A valid agent profile meets all the requirements and conflicts clauses of all service messages that the user collects. Rules from Figure 1 define agent's profile construction from service messages requirements. By Empty Profile, a user profile can be empty (base case); by Message Insertion, the elements in an installation profile are messages; by Requirement Insertion, a profile can be extended by satisfied service requirements; by Profile Extension, if a message holds in an empty profile, it can be added to an existing profile.

3.2. Rules for message construction

The operational rules in Figure 2 formulate compositionality of messages.

The rule *Atom* establishes valid content within a user profile and across other profiles with satisfied requirements. $\perp$ formulates access to contradictory messages, in which case the profile must be consistent with the negated access. $\wedge$-I allows message composition from distinct profiles; by $\wedge$-E, each composing message can be obtained from the combined profiles (with $I = \{A, B\}$). $\vee$-I says that a combined profile can access any message produced from each of the composing profiles; by the elimination $\vee$-E, each message consistently inferred by each individual profile can also be executed under the extended profile. $\rightarrow$-Introduction expresses inference of a message from a combined profile as inference between messages (Deduction Theorem); its elimination $\rightarrow$-E allows to recover such inference as profile extension (Modus Ponens).

3.3. Access Rules

In Figure 3 we present the access rules on messages. These allow a user's profile to act on messages from a distinct agent.

$\neg$-distribution expresses profile consistency: if a user profile does not allow inferring a message, then it allows inferring any other message that has no requirements including it. $read$ says that from any consistent profile a message can be read provided its requirements are satisfied (if any). $trust$ works as an elimination rule for $read$: it says that if a message is received and it preserves profile consistency, then it can be trusted. $write$ works as an elimination rule for $trust$: it says that a readable and trustable message can be sent over the network. $exec$ says that every message that is safely installed in a consistent profile holds in it.

The Introduction rule for mistrust MTrust-I says that currently held message conflicting with a newly arrived message are mistrusted; the corresponding MTrust-E allows to trust any message which is consistent with the conflict resolution by removal of the mistrusted message in the installation profile, including any required dependency, as expressed by the side condition that requires checking for any other agent. *mistrust* is a flag for facilitating removal of messages present in the user profile conflicting in view of incoming new information.

We can now offer a more general interpretation of the derivability relation $\vdash_{\mathtt{s}}$ as access and execution of some message under a given user profile:

Definition 6. *A formula* $\Gamma^{v_i} \vdash_{\mathtt{s}} RES$ *says that a message from some user* v_k *is validly accessed* $(mode(\phi^{v_k}))$ *and eventually inferred* (ϕ^{v_k}) *within a user profile with messages held by user* v_i.

4. Opportunistic Forwarding

4.1. Handshaking Protocol

In the following we present an example derivation mimicking the handshaking protocol. Here Service 1 identifies the set of messages for this protocol.

By Hello Message, a user v_i with a well-defined profile with a 'hello' message in its recognition service sends the message to the network; a user v_k reading the message and assuming it preserves consistency (e.g. there is no instruction in its profile to ingore messages from v_i), accepts it and forwards it further, including a 'hello' back to v_i.

4.2. Recipient Selection Protocol

In the following we present an example derivation mimicking the recipient selection protocol.

4.3. Message Passing Protocol

In the following we present an example derivation mimicking the message passing protocol (without mistrust). Here Service 2 is some service of any kind.

By the first premise in MP, the Handshaking Protocol is guaranteed terminating, including the Recipient Selection protocol if required; v_k then reads a message issued by v_i, checks for validity in its own profile through an application of $trust$, and if this check is passed the message is forwarded.

5. Reputation Model

In this section we illustrate the definition of the order relation $\prec$ to formalise the reputation model across agents. Higher reputation is modeled by feedback aggregation. Our

$$\frac{}{\{\} : profile}\ \text{Empty Profile} \qquad \frac{\phi^{v_j}_{i,k} : \mathcal{M}^{v_j}}{\phi^{v_j}_{i,k} : profile}\ \text{Message Insertion}$$

$$\frac{\Gamma^{v_j}, \phi^{v_j}_{i,k} : profile \qquad \Gamma^{v_j}, \phi^{v_j}_{i,k} \vdash_{\mathtt{s}} \psi^{v_k}_{j,l}}{\Gamma^{v_j}, \phi^{v_j}_{i,k} < \psi^{v_k}_{j,l} : profile}\ \text{Requirement Insertion}$$

$$\frac{\Gamma^{v_i} : profile \qquad \vdash_{\mathtt{s}} \psi^{v_k}_{j,l}}{\Gamma^{v_i}; \psi^{v_k}_{j,l} : profile}\ \text{Profile Extension}$$

Figure 1. The System `(un)SecureND`: Profile Construction Rules

$$\frac{\Gamma^{v_i}; \Gamma^{v_j} : profile}{\Gamma^{v_i}; \Gamma^{v_j} \vdash_{\mathtt{s}} \psi^{v_j}_{i,l}}\ \text{Atom, for any } \psi^{v_j}_{i,l} \in \Gamma^{v_j} \qquad \frac{\Gamma^{v_i} \vdash_{\mathtt{s}} RES \to \bot}{\Gamma^{v_i} \vdash_{\mathtt{s+1}} \neg RES}\ \bot$$

$$\frac{\Gamma^{v_i} \vdash_{\mathtt{s}} \phi^{v_i}_{i,l} \qquad \Gamma^{v_j} \vdash_{\mathtt{s'}} \psi^{v_j}_{i,m}}{\Gamma^{v_i}; \Gamma^{v_j} \vdash_{\mathtt{max(s,s')+1}} \phi^{v_i}_{i,l} \wedge \psi^{v_j}_{i,m}}\ \wedge\text{-I} \qquad \frac{\Gamma^{v_i}; \Gamma^{v_j} \vdash_{\mathtt{s}} \phi^{v_i}_{i,l} \wedge \psi^{v_j}_{i,m}}{\Gamma^{v_i}; \Gamma^{v_j} \vdash_{\mathtt{s+1}} \phi/\psi^{v_{i/j}}_{i,l/m}}\ \wedge\text{-E}$$

$$\frac{\Gamma^{v_i}; \Gamma^{v_j} \vdash_{\mathtt{s}} \phi^{v_{i/j}}_{i,l/m}}{\Gamma^{v_i}; \Gamma^{v_j} \vdash_{\mathtt{s+1}} \phi^{v_i}_{i,l} \vee \psi^{v_j}_{i,m}}\ \vee\text{-I} \qquad \frac{\Gamma^{v_i}; \Gamma^{v_j} \vdash_{\mathtt{s}} \phi^{v_i}_{i,l} \vee \psi^{v_j}_{i,m} \qquad \psi^{v_{i/j}}_{i,l/m} \vdash_{\mathtt{s'}} \xi^{v_{i/j}}_{k,n}}{\Gamma^{v_i}; \Gamma^{v_j} \vdash_{\mathtt{s'+1}} \xi^{v_{i/j}}_{k,n}}\ \vee\text{-E}$$

$$\frac{\Gamma^{v_i}; \phi^{v_i}_{i,l} \vdash_{\mathtt{s}} \psi^{v_j}_{i,m}}{\Gamma^{v_i} \vdash_{\mathtt{s+1}} \phi^{v_i}_{i,l} \to \psi^{v_j}_{i,m}}\ \to\text{-I} \qquad \frac{\Gamma^{v_i} \vdash_{\mathtt{s}} \phi^{v_i}_{i,l} \to \psi^{v_j}_{i,m} \qquad \Gamma^{v_i} \vdash_{\mathtt{s'}} \phi^{v_i}_{i,l}}{\Gamma^{v_i}; \phi^{v_i}_{i,l} \vdash_{\mathtt{s'+1}} \psi^{v_j}_{i,m}}\ \to\text{-E}$$

Figure 2. The System `(un)SecureND`: Operational Rules

system integrates the elements of the main feedback 6-tuple function from [**?**]. In particular, time is mimicked directly by derivation steps (and not accounted in the present model); context is embedded by the user profile; service and characteristics are modelled by messages. To model the set of feedback that a given agent provides with respect to a given message related to a service and characteristic, we will have to collect all formulas following receiving a message:

Definition 7 (Feedback Set)**.** *The feedback set of agent v_j for a message $\phi^{v_i}_{i,j}$, for all $v_j, v_i \in \mathcal{A}$ is the set of formulas $\psi^{v_j}_{i,k}$ such that they agree with $\phi^{v_i}_{i,j}$ for the service identifier i and are obtained by a derivation construed by a $read$ rule followed by a $\to I$ rule, i.e.*

$$FS^{v_j}(\phi^{v_i}_{i,j}) = \{\psi^{v_j}_{i,k} \mid \Gamma^{v_j} \vdash_{\mathtt{s}} Read(\phi^{v_i}_{i,j}) \to \psi^{v_j}_{i,k}\}$$

By way of example, consider the following simple derivation, which induces $FS^{v_k}(m^{v_{i,j}}_{2,1}) = \{m^{v_k}_{2,2}\}$:

Notice that by construction this set includes only feedback to received messages that are consistent with the current user's profile.

Definition 8 (Agent's Perception)**.** *The perception of agent v_j for a message $\phi^{v_i}_{i,j}$, for all $v_j, v_i \in \mathcal{A}$ is the sum of elements of the feedback set over that formula, weighted by the step of the derivation at which it is obtained:*

$$AP^{v_j}(\phi^{v_i}_{i,j}) = \sum_{FS^{v_i}(\phi^{v_j}_{i,k})} (\mathtt{s}(\psi^{v_j}_{i,k} \in FS^{v_i}(\phi^{v_j}_{i,k})))$$

Intuitively, the value of $\mathtt{s}$ at each step of each derivation leading to each formula in the feedback set of an agent to a given service and characteristic is summed up to provide a value that increases linearly to reflect a step value for a time function. The value of $AP^{v_j}(\phi^{v_i}_{i,j})$ will reflect the aggregation of all the feedback provided on each characteristics of a given service.

We can now generalize to the set of all feedback on a characteristic for a given service, remembering that these are given in a pre-order so that the position of the characteristic in that order is mapped into an integer:

Definition 9 (Agent's Perception of Characteristic Set)**.** *The perception of agent v_j for a set of messages $\mathcal{M}^{\mathcal{A}}_{S_i}$ from agents in $\mathcal{A}$ about service S_i is the sum of elements of the feedback set over the messages received about that service, weighted by the steps of the derivation at which it is obtained and further by the value $\mathtt{r}(k)$ of the rank of characteristic k:*

$$AP^{v_j}(\mathcal{M}^{\mathcal{A}}_{S_i}) = \sum_{FS^{v_i}(\phi^{v_j}_{i,k} \dots \phi^{v_n}_{i,k})} (1 - \mathtt{r}(k)(\mathtt{s}(\psi^{v_j}_{i,k} \in FS^{v_i}(\phi^{v_j}_{i,k} \dots \phi^{v_n}_{i,k}))))$$

$$\frac{\Gamma^{v_i} \vdash_{\mathtt{s}} \neg mode(\psi^{v_j}_{i,l})}{\Gamma^{v_i} \vdash_{\mathtt{s+1}} mode(\neg\psi^{v_j}_{i,l})}\ \neg\text{-distribution} \qquad \frac{}{\Gamma^{v_i} \vdash_{\mathtt{s}} Read(\psi^{v_j}_{i,l})}\ \mathit{read}$$

$$\frac{\Gamma^{v_i} \vdash_{\mathtt{s}} Read(\psi^{v_j}_{i,l}) \qquad \Gamma^{v_i};\psi^{v_j}_{i,l} : profile}{\Gamma^{v_i} \vdash_{\mathtt{s+1}} Trust(\psi^{v_j}_{i,l})}\ \mathit{trust}$$

$$\frac{\Gamma^{v_i} \vdash_{\mathtt{s}} Read(\psi^{v_j}_{i,l}) \qquad \Gamma^{v_i} \vdash_{\mathtt{s'}} Trust(\psi^{v_j}_{i,l})}{\Gamma^{v_i} \vdash_{\mathtt{s'+1}} Write(\psi^{v_j}_{i,l})}\ \mathit{write} \qquad \frac{\Gamma^{v_i} \vdash_{\mathtt{s}} Write(\psi^{v_j}_{i,l})}{\Gamma^{v_i} \vdash_{\mathtt{s+1}} \psi^{v_j}_{i,l}}\ \mathit{exec}$$

$$\frac{\Gamma^{v_i} \vdash_{\mathtt{s}} Read(\psi^{v_j}_{i,l}) \rightarrow \bot \qquad \Gamma^{v_i} \setminus \{\neg\psi^{v_i}_{i,l}\} : profile}{\Gamma^{v_i} \setminus \{\neg\psi^{v_i}_{i,l}\} \vdash_{\mathtt{s+1}} \neg Trust(\neg\psi^{v_i}_{i,l})}\ \text{MTrust-I}$$

$$\frac{\Gamma^{v_i} \setminus \{\neg\psi^{v_i}_{i,l}\} \vdash_{\mathtt{s}} \neg Trust(\neg\psi^{v_i}_{i,l}) \qquad \Gamma^{v_k};\psi^{v_j}_{i,j} : profile}{\Gamma^{v_i} \setminus \{\neg\psi^{v_i}_{i,l}\};\Gamma^{v_k} \vdash_{\mathtt{s+1}} Trust(\psi^{v_j}_{i,l})}\ \text{MTrust-E, } \forall v_k \prec v_j$$

Figure 3. The System `(un)SecureND`: Access Rules

$$\frac{\Gamma^{v_i} : profile \qquad \Gamma^{v_i} \vdash_{\mathtt{1}} hello^{v_i}_{1,1}}{\Gamma^{v_i} \vdash_{\mathtt{2}} Write(hello^{v_i}_{1,1})}\ \text{Hello Message}$$

$$\frac{\Gamma^{v_i} \vdash_{\mathtt{1}} Write(hello^{v_i}_{1,1}) \qquad \Gamma^{v_k} \vdash_{\mathtt{2}} Read(hello^{v_i}_{1,1}) \qquad \Gamma^{v_k}; hello^{v_i}_{1,1} : profile}{\Gamma^{v_k}; hello^{v_i}_{1,1} \vdash_{\mathtt{3}} Write(hello^{v_k}_{1,1})}\ \text{Response Message}$$

Figure 4. The Handshaking Protocol

$$\frac{\Gamma^{v_k}; hello^{v_i}_{1,1} \vdash_{\mathtt{1}} Write(hello^{v_k}_{1,1}) \qquad \Gamma^{v_j}; hello^{v_i}_{1,1} \vdash_{\mathtt{2}} Write(hello^{v_j}_{1,1}) \qquad v_k \prec v_j}{\Gamma^{v_i};\Gamma^{v_k} : profile}\ \text{Recipient Selection}$$

Figure 5. The Handshaking Protocol

$$\frac{\dfrac{\dfrac{\Gamma^{v_i};\Gamma^{v_k} : profile \qquad \Gamma^{v_i} \vdash_{\mathtt{1}} Write(m^{v_i}_{2,1})}{\Gamma^{v_k} \vdash_{\mathtt{2}} Read(m^{v_i}_{2,1})}\ \text{MP} \qquad \Gamma^{v_k}; m^{v_i}_{2,1} : profile}{\Gamma^{v_k} \vdash_{\mathtt{3}} Trust(m^{v_i}_{2,1})}}{\Gamma^{v_k} \vdash_{\mathtt{4}} Write(m^{v_i}_{2,1})}$$

Figure 6. The Message Passing Protocol

Using the agent's perception of characteristic set, we can define the order of reputation with respect to services, which establishes a higher position for the agent whose perception on the characteristics set for that Service is greater.

Definition 10 (Reputation)**.** $\forall v_i, v_j \in \mathcal{V}, S_i \in \mathcal{S}, v_i \prec v_j : S_i \leftrightarrow AP^{v_i}(\mathcal{M}^{\mathcal{A}}_{S_i}) > AP^{v_j}(\mathcal{M}^{\mathcal{A}}_{S_i})$.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{\Gamma^{v_i};\Gamma^{v_k} : profile \qquad \Gamma^{v_j};\Gamma^{v_k} : profile}{\Gamma^{v_i};\Gamma^{v_j};\Gamma^{v_k} : profile}
        \qquad \Gamma^{v_k} \vdash_1 Write(m_{2,1}^{v_{i,j}})
      }{\Gamma^{v_k} \vdash_2 Read(m_{2,1}^{v_{i,j}})}
      \qquad \Gamma^{v_k}; m_{2,1}^{v_{i,j}} : profile
    }{\Gamma^{v_k} \vdash_3 Trust(m_{2,1}^{v_{i,j}})}
  }{\Gamma^{v_k} \vdash_4 Write(m_{2,1}^{v_{i,j}})}
  \qquad \Gamma^{v_k}; m_{2,1}^{v_{i,j}} \vdash_5 m_{2,2}^{v_k}
}{\Gamma^{v_k} \vdash_6 m_{2,1}^{v_i} \rightarrow m_{2,2}^{v_k}}
$$

Figure 7. An Example Feedback Set