A Proof-theoretic Trust and Reputation Model for VANET

Giuseppe Primiero, Franco Raimondi, Taolue Chen, Rajagopal Nagarajan
Department of Computer Science
Middlesex University London
United Kingdom
Email: G.Primiero—F.Raimondi—T.Chen—R.Nagarajan@mdx.ac.uk

***Abstract*—Vehicle Ad Hoc Networks (VANETs) 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. Formal correctness, resolution of contradictions and proven safety of transitive operations within the infrastructure remain currently 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 which guarantees consistency of the messaging protocol and security of transitive transmissions.**

1. Introduction

Vehicle Ad Hoc Networks (VANETs) consist of vehicles and roadside units networks created to enhance transportation systems through V2V and V2I communications. Due to their distributed and dynamic nature, such networks are open to several types of threats, including false message propagation. Trust and reputation are among the concept most used to ensure integrity, reliability and safety of services. Several methods have been implemented, see [8] for a recent overview. Trust models in VANETs differ especially for the main object of the model: entity-centric [3], [1], data-centric [6], [2] and combined [10]. Among the models that combine trust and reputation, [9] offers an analysis that accounts for reputation as a characteristic of message forwarding, vehicles, drivers and other agents as well: here reputation is therefore based on a descriptive ontology of the model and is used to feedback in the system.

Unfortunately, none of the current models seem to have focused on a formal correctness requirement to ensure that derivability of messages passing protocol is checkable. Moreover, an additional problem seems to have been ignored, namely how to ensure that safety is preserved over transitive message passing operations. This paper provides a solution to both problems above. We formulate a proof-theoretic translation of the trust and reputation model for VANET offered in [9] in an extension of the natural deduction calculus `(un)SecureND` from [4]. The aim is, first of all, to show that the trust properties instantiated through our calculus faithfully reflect those in a VANET network; accordingly, we show how non-trustworthy interactions can be proven to be such through a proof-checking method. On a higher level, the model offered by `(un)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. Finally, thanks to the structural properties of our calculus, we consider a transitive messgae passign operation in the form of instances of a cut rule, whose normalization guarantees safety.

2. `(un)SecureND`

(un)`SecureND` is a natural deduction calculus defining trust, mistrust and distrust protocols introduced in [5] for the positive fragment and in [4] 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`)**.**

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

2.1. 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 service and characteristic identifiers, so that: $\phi^{v_i}_{i,j}$ says that message ϕ about service S_i and characteristic C_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 $\phi^{v_i}_{S_i,C_j}$ is usually abbreviated as $\phi^{v_i}_{i,j}$. $mode$ 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 about service i and characteristic k signed from agent v_j 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 ϕ about service i and characteristic k signed from agent v_j holds for any agent's profile at step $\mathtt{s}$.*

Messages satisfy a ranking based on that of characteristics:

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

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.

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.

2.2. 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).

2.3. 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 rule MTrust-I says that currently held message conflicting with a newly arrived message are mistrusted, i.e. removed from the current profile until none of its consequences are included; the corresponding MTrust-E elimination allows to trust any message consistent with the conflict resolution by removal of the mistrusted message in the user profile, including any required dependency, as expressed by the side condition that requires checking with any other agent who has higher reputation than the sender of the original message. *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 5. *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 at step $\mathtt{s}$.*

2.4. Structural Rules

Structural rules hold with restrictions for (un)`SecureND`, see Figure 4. As a result the system qualifies as substructural, see e.g. [7]. Weakening is constrained by an instance of $trust$: it says valid information is preserved under vehicle's profile extension, assuming the latter is provably consistent and therefore no refresh is required. Contraction is constrained by preservation of ordering: it says that removing identical messages from a vehicle's profile is admissible, with the constraint that the copy from the agent with higher reputation is preserved. Exchange is doubly constrained by order: it says that reorder of messages is admissible if there is no involved dependency between them. Finally, the Cut rule expresses validity under profile extension: if a message $\phi_{i,j}$ is validly for agent v_i and after messaging it to v_j the latter can infer $\phi_{i,k}$, then v_i can infer $\phi_{i,k}$ by setting a message protocol with v_j.

The cut rule justifies the following result:

Theorem 1 (Normalization). *Any* (un)`SecureND` *derivation with an occurrence c of the Cut rule can be transformed* *into another derivation with the same end sequent without c using only trust.*

Proof. By induction on the derivation D which is the redex of the cut-elimination. Assuming c is the only Cut rule and it is the last inference rule of the redex, the derivation D' which is the contractum of the cut-elimination contains a descendent of the cut obtained by an instance of Weakening under trust. Because the formula obtained by the cut is, by hypothesis, derivable from the weaker protocol, it will also be derivable from the weaker and the stronger protocol together. When c is not the last inference rule of the redex, then the descendent of the cut will admit all similar Weakenings preserving the one occurring in the cut; those imports by Weakening will occur also in the contractum of the cut rule and can be traced back up to the one formulation of the import that occurs in the cut rule. □

Normalization justifies a safety property of our trust and reputation model over transitive transmissions: for each vehicle v_i, v_j, v_k, if v_k holds information $\phi_{i,j}$ and this information is passed to v_j, then every valid message derived from $\phi_{i,j}$ by v_k can be inferred by v_j assuming the consistency (by trust) of its profile with that of v_k; similarly now, v_j can pass $\phi_{i,j}$ to v_i, and the latter can infer from there, assuming its profile is consistent with those of v_j, v_k.

3. Opportunistic Forwarding

In Figure 5 we present an example derivation mimicking an 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.

In Figure 6, we present an example derivation mimicking the recipient selection protocol. Here the idea is as follows: after v_i broadcasts a 'hello' message, v_k, v_j both receive and accept the message; at this stage a recipient is selected on the basis of the reputation order between v_k and v_j, so that a new profile is built out of v_i and the higher of the two recipients, thus mimicking a communication channel.

In Figure 7, 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.

4. Reputation Model

In this section we illustrate the definition of the order relation $\prec$ to formalise the reputation model across agents. Higher reputation is modelled by feedback aggregation. Our system integrates the elements of the main feedback 6-tuple function from [9]. In particular, time is mimicked directly by derivation steps; 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 6 (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 $\rightarrow I$ rule, i.e.*

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

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 7 (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 8 (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 $\mathtt{r}(C_k)$ of the rank of characteristic k:*

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

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

Definition 9 (Reputation)**.** $\forall v_i, v_j \in \mathcal{V}, S_i \in \mathcal{S}, v_i \prec v_j \leftrightarrow AP^{v_i}(\mathcal{M}^{\mathcal{A}}_{S_i,C_k}) > AP^{v_j}(\mathcal{M}^{\mathcal{A}}_{S_i,C_k})$.

5. Conclusions

In this paper we have formulated a proof-theory for trust and reputation in VANETs. Our language is modelled on the logic `(un)SecureND`, including an explicit $trust$ function on formulas to guarantee consistency check at each retrieval step (after a $read$ function), before forwarding is granted for a package (by a $write$ function). Forwarding is modelled in an opportunistic fashion, selecting receivers on the basis of their reputation ranking. Trust on forwarding also guarantees correctness on transitive transmissions. Moreover, reputation is used to implement the resolution protocol for restoring information after removing previously stored data. Validation of the system is obtained by implementation of the `(un)SecureND` calculus as a large inductive type in the Coq proof assistant. The development is available at https://github.com/gprimiero/SecureNDC. A characteristic of the logic `(un)SecureND` is its substructural nature, which in future work can be exploited to investigate cases of strengthened and limited resource redundancy for fault tolerance and source shuffling for security. Other applications of negative trust can be investigated to distinguish between malevolent and simply unsuccessful sources.

References

[1] Félix Gómez Mármol and Gregorio Martínez Pérez. TRIP, a Trust and Reputation Infrastructure-based Proposal for Vehicular Ad Hoc Networks. *J. Netw. Comput. Appl.*, 35(3):934–941, May 2012.

[2] Nai-Wei Lo and Hsiao-Chien Tsai. A Reputation System for Traffic Safety Event on Vehicular Ad Hoc Networks. *EURASIP Journal on Wireless Communications and Networking*, 2009(1):125348, 2009.

[3] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen. A Multifaceted Approach to Modeling Agent Trust for Effective Communication in the Application of Mobile Ad Hoc Vehicular Networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(3):407–420, May 2011.

[4] Giuseppe Primiero. A Calculus for Distrust and Mistrust. In Sheikh Mahbub Habib, Julita Vassileva, Sjouke Mauw, and Max Mühlhäuser, editors, *Trust Management X - 10th IFIP WG 11.11 International Conference, IFIPTM 2016, Darmstadt, Germany, July 18-22, 2016, Proceedings*, volume 473 of *IFIP Advances in Information and Communication Technology*, pages 183–190. Springer, 2016.

[5] Giuseppe Primiero and Franco Raimondi. A typed natural deduction calculus to reason about secure trust. In Ali Miri, Urs Hengartner, Nen-Fu Huang, Audun Jøsang, and Joaquín García-Alfaro, editors, *2014 Twelfth Annual International Conference on Privacy, Security and Trust, Toronto, ON, Canada, July 23-24, 2014*, pages 379–382. IEEE, 2014.

[6] Maxim Raya, Panagiotis Papadimitratos, Virgil D. Gligor, and Jean-Pierre Hubaux. On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks. In *INFOCOM*, pages 1238–1246. IEEE, 2008.

[7] Greg Restall. *An Introduction to Substructural Logics*.

[8] Seyed Ahmad Soleymani, Abdul Hanan Abdullah, Wan Haslina Hassan, Mohammad Hossein Anisi, Shidrokh Goudarzi, Mir Ali Rezazadeh Baee, and Satria Mandala. Trust management in vehicular ad hoc network: a systematic review. *EURASIP Journal on Wireless Communications and Networking*, 2015(1):146, 2015.

[9] R Vanni, L.M.S. Jaimes, G. Mapp, and E. Moreira. Ontology Driven Reputation Model for VANET. In *AICT 2016, The Twelfth Advanced International Conference on Telecommunications* , pages 14–19. IARIA, 2016.

[10] Yu-Chih Wei and Yi-Ming Chen. *Reliability and Efficiency Improvement for Trust Management Model in VANETs*, pages 105–112. Springer Netherlands, Dordrecht, 2012.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{\Gamma^{v_i} \vdash_{\mathtt{s}} \phi^{v_i}_{i,j} \qquad \Gamma^{v_i} \vdash_{\mathtt{s'}} Trust(\phi^{v_j}_{j,k})}{\Gamma^{v_i}; \phi^{v_j}_{j,k} \vdash_{\mathtt{max(s,s'+1)}} \phi^{v_i}_{i,j}}\text{ Weakening} \quad \frac{\Gamma^{v_i}; \phi^{v_j}_{j,k}; \phi^{v_k}_{j,k} \vdash_{\mathtt{s}} \psi^{v_i}_{i,j} \qquad v_j \prec v_k}{\Gamma^{v_i}; \phi^{v_j}_{j,k} \vdash_{\mathtt{s+1}} \psi^{v_i}_{i,j}}\text{ Contraction}$$

$$\frac{\Gamma^{v_i}; \phi^{v_i}_{i,j}; \phi^{v_i}_{i,k} \vdash_{\mathtt{s}} \psi^{v_i}_{i,j} \qquad \phi^{v_i}_{i,j} \not\prec \phi^{v_i}_{i,k}}{\Gamma^{v_i}; \phi^{v_i}_{i,k}; \phi^{v_i}_{i,j} \vdash_{\mathtt{s+1}} \psi^{v_i}_{i,j}}\text{ Profile Exchange}$$

$$\frac{\Gamma^{v_i} \vdash_{\mathtt{s}} \phi^{v_i}_{i,j} \qquad \Gamma^{v_j}, \phi^{v_i}_{i,j} \vdash_{\mathtt{s'}} \phi^{v_j}_{i,k}}{\Gamma^{v_i}; \Gamma^{v_j} \vdash_{\mathtt{max(s,s')+1}} \phi^{v_j}_{i,k}}\text{ Cut}$$

Figure 4. The System `(un)SecureND`: Structural Rules

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

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

Figure 5. The Handshaking Protocol

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

Figure 6. The Handshaking Protocol

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

Figure 7. The Message Passing Protocol

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\Gamma^{v_i}; \Gamma^{v_k} : profile \qquad \Gamma^{v_j}; \Gamma^{v_k} : profile}{\Gamma^{v_i}; \Gamma^{v_j}; \Gamma^{v_k} : profile} \qquad \Gamma^{v_k} \vdash_{1} Write(m^{v_{i,j}}_{2,1})}{\Gamma^{v_k} \vdash_{2} Read(m^{v_{i,j}}_{2,1})} \qquad \Gamma^{v_k}; m^{v_{i,j}}_{2,1} : profile}{\Gamma^{v_k} \vdash_{3} Trust(m^{v_{i,j}}_{2,1})}}{\Gamma^{v_k} \vdash_{4} Write(m^{v_{i,j}}_{2,1})} \qquad \Gamma^{v_k}; m^{v_{i,j}}_{2,1} \vdash_{5} m^{v_k}_{2,2}}{\Gamma^{v_k} \vdash_{6} m^{v_i}_{2,1} \to m^{v_k}_{2,2}}$$

Figure 8. An Example Feedback Set