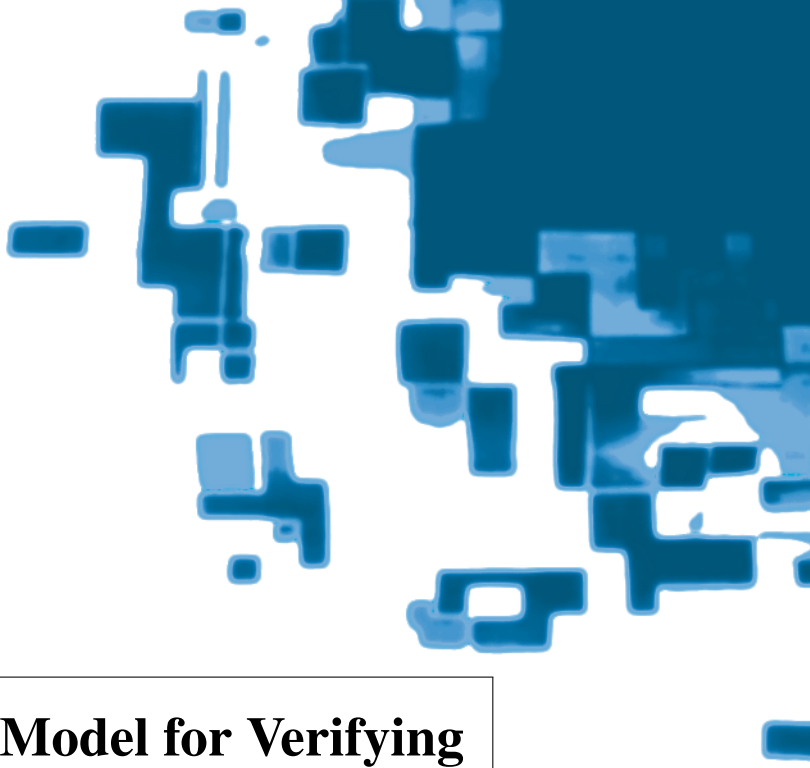


A More Realistic Model for Verifying Route Validity in Ad-Hoc Networks

Ali Kassem, Pascal Lafourcade, Yassine Lakhnech

Verimag Research Report n^o TR-2013-10

October 28, 2013

Unité Mixte de Recherche 5104 CNRS - Grenoble INP - UJF
Centre Équation
2, avenue de VIGNATE
F-38610 GIERES
tel : +33 456 52 03 40
fax : +33 456 52 03 50
http://www-verimag.imag.fr

A More Realistic Model for Verifying Route Validity in Ad-Hoc Networks

Ali Kassem, Pascal Lafourcade, Yassine Lakhnech

Université Grenoble 1, CNRS, VERIMAG, France
firstname.lastname@imag.fr

October 28, 2013

Abstract

Many cryptographic protocols aim at ensuring the *route validity* in ad-hoc networks, *i.e.* the established route representing an exists path in the network. However, flaws have been found in some protocols that are claimed secure (*e.g.* the attack on SRP applied to DSR). Some formal models and reduction proofs have been proposed to give more guarantees when verifying route validity and facilitate verification process. The existing approaches assume the cooperative attacker model. In this paper, we consider the non-cooperative attacker model, and we show that verifying the route validity under the non-cooperative model requires to verify only five topologies, each containing four nodes, and to consider only three malicious (compromised) nodes. Furthermore, we prove that a protocol is secure for any topology under the non-cooperative model, if and only if, it is secure for any topology under the cooperative model.

Keywords: Routing protocols, non-cooperative attacker, route validity, reduction proof.

Reviewers: Pascal Lafourcade

How to cite this report:

```
@techreport {TR-2013-10,
  title = {A More Realistic Model for Verifying Route Validity in Ad-Hoc Networks},
  author = {Ali Kassem, Pascal Lafourcade, Yassine Lakhnech},
  institution = {{Verimag} Research Report},
  number = {TR-2013-10},
  year = {}
}
```

Contents

1 Introduction

Wireless ad-hoc networks have no existing infrastructure. This enables them to play more and more important role in extending the coverage of traditional wireless infrastructure (*e.g.* cellular networks, wireless LAN, etc.). These networks have no central administration control, and thus the presence of dynamic and adaptive routing protocols is necessary for them to work properly. Routing protocols aim to establish a route between distant nodes, enabling wireless nodes to communicate with the nodes that are outside their transmission range. Attacking routing protocol may disable the whole network operation. For example, forcing two nodes to believe in an invalid route (a path that is not in the network) will prevent them from communicating with each other. Several routing protocols [13, 8, 14] have been proposed to provide more guarantees on the resulting routes for ad-hoc networks. However, they may be still subject to attacks. For example, a flaw has been discovered on the Secure Routing Protocol SRP [13] when it applied to Dynamic Source Routing protocol DSR [9], allowing an attacker to modify the route, which makes the source node accept an invalid one [4]. Another attack was found on the Ariadne protocol [8] in the same paper. This shows that designing secure routing protocol is a difficult and error-prone task. An NP-decision procedure has been proposed by M. Arnaud *et al.* [2] for analysing routing protocols looking for attacks on route validity in case of a fixed topology. However, the existence of an attack strongly depends on the network topology, *i.e.* how nodes are connected and where malicious nodes are located. This results in an infinite number of topologies to verify, which is not tractable. Indeed, in contrast to classical Dolev-Yao attacker [7] that controls all the communications, an attacker for routing protocols has to situate somewhere in the network. It can control only a finite number of nodes (typically one or two), and thus it can listen to the communication of its neighbours but it is not possible to listen beyond the neighbouring nodes. Cortier *et al.* [6] proposed a reduction proof when looking for route validity property under the *cooperative attacker model*, *i.e.* a model that allows distant malicious nodes to communicate using out-of-band resources, and thus to share their knowledge.

In fact, due to their minimal configuration and quick deployment ad-hoc networks are suitable for emergency situations like natural disasters or military conflicts where no infrastructure is available. So, usually it is difficult to have common channels between malicious nodes. As an example, consider the case of ad-hoc sensors that are thrown from a plane into the enemy field during a battle. Also, in-band-communications between malicious nodes are unfeasible in some cases where nodes have low power capabilities (*e.g.* sensor networks). Moreover, it is well-known that the presence of several colluding malicious nodes often yields

straightforward attacks [5, 10].

Contributions: We consider route validity property under the *non-cooperative* attacker model, where malicious nodes work independently, *i.e.* they have no ability to share their knowledge. We use the CBS♯ [12] calculus to model routing protocol, and the transition rules introduced in [2] to model the communications between nodes, after updating them to handle the behaviour of the non-cooperative malicious nodes instead of the cooperative ones.

Then, we revisit the work presented by Cortier *et al.* in [6], where they show that when looking for attacks on route validity under the cooperative model it is enough to check only five particular small topologies. We show that the same result is also valid in case the of non-cooperative model: first, we show that if there is an attack on a routing protocol in a certain topology under the non-cooperative model, then there is an attack on this protocol in a smaller topology obtained from the original one by a simple reduction. Then, we show that applying the reduction procedure to any topology leads to (at most) five small topologies. The resulting topologies are the same ones obtained in [6] under the cooperative model.

Finally, we prove that a protocol is secure under the cooperative model in any topology, if and only if, it is secure under the non-cooperative model in any topology. The latter result does not hold when we consider only one fixed topology.

Related work: The non-cooperative model is already used in [3] to analyse the web-service applications looking for attacks that exploit XML format. Verifying route validity under this model is equivalent to satisfiability of general constraints where knowledge monotonicity does not hold. The satisfiability of such kind of constraints has been proven to be NP-complete [11, 3]. However, verifying routing protocols requires considering an infinite number of topologies as we mentioned before.

To the best of our knowledge, the first approach proposing a reduction result in the context of routing protocols is [1]. In this paper, the authors have shown how to reduce the number of network topologies that need to be considered, taking advantage of the symmetries. However, the total number of networks is still large, for example, 5698 networks need to be considered when the number of nodes is six.

Our work follows the spirit of [6] where it has been shown that only five topologies need to be considered when looking for attacks on properties such as route validity under the cooperative model. Our work differs by considering the non-cooperative model which is a weaker one. We show that the problem of checking if a certain protocol is secure for any topology is equivalent under the two models. However, in a fixed topology we may find a protocol that is secure under the non-cooperative model, but not under the cooperative one. Actually, considering a powerful attacker model by giving malicious nodes the ability to share their knowledge may introduce some false positive attacks in the sense that we may found some attacks that can not be to mounted in practice. Also, we should note that studying protocol security under the non-cooperative model requires strictly less executions to be considered.

Outline: We introduce notations and attacker capabilities in Section 2. Then in Section 3, we show how to model routing protocols by process calculus, and we define the security property. In Section 4, we present our reduction proof and show that only five topologies are sufficient. Finally, before concluding, we make a comparison between the cooperative and non-cooperative models in Section 5.

2 Preliminaries

To model messages we consider an arbitrary term algebra and deduction system.

2.1 Messages

We use *terms* to represent messages and *function symbols* to represent cryptographic primitives such as encryption and hash function. We consider a *signature* $(\Sigma, \mathbb{S})$ made of a set of sorts $\mathbb{S}$ and a set of function symbols Σ with *arities*, $ar(\cdot) : \Sigma \mapsto \mathbb{N}$. The set of function symbols of arity n is denoted by Σ_n. For a function symbol $f \in \Sigma_n$ we have that $f : s_1 \times \cdots \times s_n \mapsto s$ with $s, s_1, \ldots, s_n \in \mathbb{S}$. We consider a countable set of variables $\mathbb{X}$. For a set $X \subseteq \mathbb{X}$, we define a set of terms $\mathbb{T}(\Sigma, X)$ to be the smallest set containing Σ_0

and X, such that for a function symbol $g \in \Sigma_n$: if $t_1, \ldots, t_n \in \mathbb{T}(\Sigma, X)$ then $g(t_1, \ldots, t_n) \in \mathbb{T}(\Sigma, X)$. In the case that $X = \emptyset$, we simply write $\mathbb{T}(\Sigma)$, this is the set of ground terms.

We assume two special sorts: the sort Agent that only contains agent's names and variables, and the sort Term that subsumes all other sorts so that any term is of the sort Term. As an example, a typical signature for representing the primitives used in SRP [13] protocol is the signature $(\Sigma_{SRP}, \mathbb{S}_{SRP})$ defined by $\mathbb{S}_{SRP} = \{\mathsf{Agent}, \mathsf{List}, \mathsf{Term}\}$ and $\Sigma_{SRP} = \{hmac.(\cdot), \langle\cdot,\cdot\rangle, ::, [], req, rep\}$, where req and rep are unitary constants identify the request and response phases respectively, [] represents an empty list and other symbols are defined as follows:

$$\langle\cdot,\cdot\rangle : \mathsf{Term} \times \mathsf{Term} \to \mathsf{Term} \qquad :: : \mathsf{Agent} \times \mathsf{List} \to \mathsf{List}$$
$$hmac.(\cdot) : \mathsf{Term} \times \mathsf{Term} \to \mathsf{Term}$$

The symbol $hmac.(\cdot)$ takes two terms and computes the message authentication code MAC of the first term with the second one as a key. The operator $\langle\cdot,\cdot\rangle$ produces a concatenation of two terms, and the operator :: is the list constructor. We write $<t_1, t_2, t_3>$ for the term $\langle\langle t_1, t_2\rangle, t_3\rangle$, and $[t_1, t_2, t_3]$ for $((] :: t_1) :: t_2) :: t_3$.

Substitutions and unifications: A *substitution* σ is a mapping from $\mathbb{X}$ to $\mathbb{T}(\Sigma, \mathbb{X})$ with the domain $dom(\sigma) = \{x \in \mathbb{X} \mid \sigma(x) \neq x\}$. We consider only *well-sorted substitutions*, that is substitution for which x and $\sigma(x)$ have the same sort. We extend σ to a homomorphism on functions, processes and terms as expected. We say that the two terms t and s are *unifiable* if there exists a substitution θ, called *unifier*, such that $\theta(t) = \theta(s)$. We define the *most general unifier* (for short mgu) of two terms t and s to be a unifier, denoted $mgu(t, s)$, such that for any unifier θ of t and s there exists a substitution σ with $\theta = \sigma \circ mgu(t, s)$ where $\circ$ is a composition of two mappings. We write $mgu(t, s) = \perp$ when t and s are not unifiable.

2.2 Attacker Capabilities

We consider a non-cooperative model where there are multiple independent attackers that have no ability to share knowledge between each other. The ability of each attacker is modelled by a deduction relation $\vdash$. Such a relation is defined through an inference system, *i.e.* a finite set of rules of the form $\frac{t_1 \cdots t_n}{t}$, where $t, t_1, \ldots, t_n \in \mathbb{T}(\Sigma, \mathbb{X})$. A term t is *deducible* from a set of terms I, denoted by $I \vdash t$, if there exists a proof tree with a root labelled by t and leaves labelled by $t' \in I$ and every intermediate node is an instance of one of the rules of the inference system. We can associate to the SRP signature $(\Sigma_{SRP}, \mathbb{S}_{SRP})$, the following inference system:

$$\frac{t_1 \quad t_2}{\langle t_1, t_2\rangle} \qquad \frac{\langle t_1, t_2\rangle}{t_i}\ i \in \{1,2\} \qquad \frac{l_1 \quad l_2}{l_1 :: l_2} \qquad \frac{l_1 :: l_2}{l_i}\ i \in \{1,2\} \qquad \frac{t_1 \quad t_2}{hmac_{t_2}(t_1)}$$

The terms t_1 and t_2 are of sort Term, l_1 is of sort List, whereas l_2 is of sort Agent. The system gives the attacker an ability to concatenate terms, build lists, as well as to retrieve their components. The last inference rule models the fact that the attacker can also compute a MAC provided he knows the corresponding key.

3 Modelling Routing Protocols

3.1 Process Calculus

The intended behaviour of each node in the network can be modelled by a process defined using the grammar given in Figure 1. We use the CBS♯ calculus introduced in [12]. We parameterized them by a set $\mathbf{P}$ of predicates to represent the checks performed by the agents, and a set $\mathcal{F}$ of functions over terms to represent the computations performed by the agents. The set of functions $\mathcal{F}$ contains functions that are more complex than basic cryptographic primitives represented by Σ, for example a function $f : (x, y, z) \mapsto hmac_z(\langle x, y\rangle)$ which takes three terms, concatenates the first two and then computes the MAC over them with the third term. They can also be used to model operations on lists, for example we can define a function that take a list and return its reverse.

$P, Q ::=$	Processes
0	null process.
$out(f(t_1, \ldots, t_n)).P$	emission
$in(t).P$	reception
$if\ \Phi\ then\ P$	conditional
$P \mid Q$	parallel composition.
$!P$	replication
$new\ m.P$	fresh name generation

where $t, t_1, \ldots, t_n$ are terms, m is a name, $f \in \mathcal{F}$ and Φ is a formula:

$\Phi, \Phi_1, \Phi_2 ::=$	Formula
$p(t'_1, \ldots, t'_n)$	$p \in \mathbf{P}$, $t'_1, \ldots, t'_n$ are terms
$\Phi_1 \wedge \Phi_2$	conjunction

Figure 1: Process grammar

The process $out(f(t_1, \ldots, t_n)).P$ first computes the term $t = f(t_1, \ldots, t_n)$, emits t, and then behaves like P. The reception process $in(t).P$ expects a message m matching the pattern t and then behaves like $\sigma(P)$ where $\sigma = mgu(m, t)$. The process $if\ \Phi\ then\ P$ tests whether Φ is true, if Φ is true it then behaves like P. Two processes P and Q running in parallel represented by the process $P|Q$. The replication process $!P$ denotes an infinite number of copies of P, all running in parallel. The process $new\ m.P$ creates a fresh name m and then behaves like P. Sometimes, for the sake of clarity we omit the null process. We assume that the predicates $p \in \mathbf{P}$ are given together with their semantics that may depend on the underlying graph G. We consider two kinds of predicates: a set $\mathbf{P_I}$ of predicates whose semantics is independent of the graph and a set $\mathbf{P_D}$ of predicates whose semantics is dependent on the graph. For a graph dependent formula Φ and a graph G, we write $[\![\Phi]\!]_G = true$ (resp. $false$) to denote that Φ is $true$ (resp. $false$) in G. For example, we can use the predicates $\mathbf{P_{SRP}} = \mathbf{P_I} \cup \mathbf{P_D}$ for SRP, with $\mathbf{P_I} = \{\mathsf{checksrc}, \mathsf{checkdest}\}$ and $\mathbf{P_D} = \{\mathsf{check}, \mathsf{checkl}\}$. The purpose of the $\mathbf{P_I}$ predicates is to model some checks that are performed by the source when it receives the route. The semantics of these predicates is defined as follows:

- $\mathsf{checksrc}(S, l) = true$ if and only if l is of sort List and its first element is S,
- $\mathsf{checkdest}(D, l) = true$ if and only if l is of sort List and its last element is D.

The predicates $\mathsf{checksrc}(S, l)$ and $\mathsf{checkdest}(D, l)$ are used by the source process to verify that the first and last nodes of the established route are the source and destination of the route discovery respectively.

While, the purpose of the $\mathbf{P_D}$ predicates is to model neighbourhood checks. Given a graph $G = (V, E)$, their semantics is defined as follows:

- $[\![\mathsf{check}(A, B)]\!]_G = true$ if and only if $(A, B) \in E$ or $(B, A) \in E$,
- $[\![\mathsf{checkl}(C, l)]\!]_G = true$ if and only if C appears in l and for any l' subterm of l we have $(A, C) \in E$ if $l' = l_1 :: A :: C$ and $(C, B) \in E$ if $l' = l_1 :: C :: B$.

The aim of the predicate $[\![\mathsf{check}(A, B)]\!]_G$ is to check if A and B are neighbours in G, while the aim of the predicate $[\![\mathsf{checkl}(C, l)]\!]_G$ is to check if the node C appears in l between two neighbours in G. We assume that each nodes knows its neighbours in the network, this can be achieved by running a certain neighbour discovery protocol in advance.

We write $fv(P)$ for the set of *free variables* that occur in P, *i.e.* the set of variables that are not in the scope of an input. We consider ground processes, *i.e.* processes P such that $fv(P) = \emptyset$, and parameterized processes, denoted $P(x_1, \ldots, x_n)$ where $x_1, \ldots, x_n$ are variables of sort Agent, and such that $fv(P) \subseteq \{x_1, \ldots, x_n\}$. A routing role is a parameterized process that does not contain any name of sort Agent. A routing protocol is then simply a set of routing roles.

The secure routing protocol SRP applied on DSR, already modelled in [6] using these process calculus. Here we give only the source process as an example. Considering the signature $(\Sigma_{SRP}, \mathbb{S}_{SRP})$ and the

predicates $\mathbf{P_{SRP}}$ introduced before, and the set $\mathcal{F}_{SRP}$ of functions over terms that only contains the identity function (omitted for sake of clarity), the process played by the source x_S initiating the search of a route towards a destination x_D is given as follows:

$$P_{src}(x_S, x_D) = \text{new } id.out(u_1).in(u_2).\text{if } \Phi_S \text{ then } 0$$

where id is a constant identifies the request, x_S, x_D are variables of sort Agent, and x_L is a variable of sort List and

$$\begin{aligned} u_1 &= \langle req, x_S, x_D, id, [] :: x_S, hmac_{k_{x_S x_D}}(\langle req, x_S, x_D, id\rangle)\rangle \\ u_2 &= \langle req, x_D, x_S, id, x_L, hmac_{k_{x_S x_D}}(\langle req, x_D, x_S, id, x_L\rangle)\rangle \\ \Phi_S &= \mathsf{checkl}(x_S, x_L) \wedge \mathsf{checksrc}(x_S, x_L) \wedge \mathsf{checkdest}(x_D, x_L) \end{aligned}$$

3.2 Configuration and Topology

Each process is located at a specified node of the network. Unlike the classical Dolev-Yao model [7], the attacker does not control the entire network but can only interact with its neighbours. More specifically, we assume that the *topology* of the network is represented by a tuple $\mathcal{T} = (G, \mathcal{M}, S, D)$ where:

- $G = (V, E)$ is an undirected graph with $V \subseteq \{A \in \Sigma_0 \mid A \text{ of sort Agent}\}$, where an edge in the graph models the fact that two agents are neighbours. We only consider graphs such that $\{(A, A) | A \in V\} \subseteq E$ which means that an agent can receive a message sent by himself;
- $\mathcal{M} = \{M_i\}_{i=1}^{i=k}$ is a set of nodes that are controlled by k attackers we have in the network, where each attacker controls only one node. Note that $\mathcal{M} \subseteq V$. These nodes that are in $\mathcal{M}$ are called malicious whereas nodes not in $\mathcal{M}$ are called honest;
- S and D are two honest nodes that represent respectively the source and the destination for which we analyse the security of the routing protocol.

Note that malicious nodes cannot communicate using out-of-band resources or hidden channels.

A *configuration* of the network is a pair $(\mathcal{P}, \mathcal{I})$ where:

- $\mathcal{P}$ is a multiset of expressions of the form $\lfloor P \rfloor_A$ that represents the process P executed by the agent $A \in V$. We write $\lfloor P \rfloor_A \cup \mathcal{P}$ instead of $\{\lfloor P \rfloor_A\} \cup \mathcal{P}$;
- We assume an independent knowledge for each attacker as we define the set of sets of terms $\mathcal{I} = \{I_i\}_{i=1}^{i=k}$, where the set I_i represents the messages seen by the malicious node $M_i \in \mathcal{M}$ as well as its initial knowledge.

A possible topology $\mathcal{T}_0 = (G_0, \mathcal{M}_0, S, D)$ is modelled in Figure 2, where M_1 and M_2 are malicious nodes (colored in black), *i.e.* $\mathcal{M}_0 = \{M_1, M_2\}$ while A is an extra honest node (colored in white). To refer to the source and destination we use $\rightarrow\!\bigcirc$ and $\bigcirc\!\rightarrow$ respectively. A typical initial configuration for the SRP protocol is

$$K = (\lfloor P_{src}(S, D) \rfloor_S \mid \lfloor P_{req}(A) \rfloor_A \mid \lfloor P_{rep}(A) \rfloor_A \mid \lfloor P_{dest}(D) \rfloor_D; \mathcal{I})$$

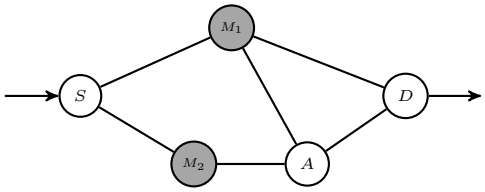


Figure 2: Topology $\mathcal{T}_0$

3.3 Execution Model

The communication system is formally defined by the rules of Figure 3. These rules are parameterized by the underlying topology $\mathcal{T} = (G, \mathcal{M}, S, D)$ with $G = (V, E)$. They are quite similar to the ones used in [2, 6] with the difference that we use an independent attackers knowledge, and we assume that the message sent by a certain malicious node can be captured by its malicious neighbours due to broadcast nature of the communications in wireless add-hoc networks, this modelled in the rule IN. The COMM rule allows nodes to communicate provided they are directly connected in the underlying graph. The exchanged message is added to the knowledge I_i of the malicious node M_i if the agent emitting the message is a direct neighbour of M_i, this reflects the fact that a malicious node can listen to the communications of its neighbours. The IN rule allows a malicious node M_i to send any message it can deduce from its knowledge I_i to one of its neighbours, and like in COMM rule this message captured by neighbour malicious nodes. The rule IF-THEN states that the node A executes the process P only if the formula Φ is true. PAR rule says that parallel processes are equivalent to parallel nodes running these processes. The replication process $!P$ expanded using the rule REPL. The last rule NEW says that nodes can use fresh names of their choice when required. The relation $\rightarrow_{\mathcal{T}}^{*}$ is the reflexive and transitive closure of $\rightarrow_{\mathcal{T}}$.

COMM $\quad (\{\lfloor in(t'_j).P_j \rfloor_{A_j} \mid mgu(t, t'_j) \neq \bot, (A, A_j) \in E\}$
$\cup \lfloor out(f(t_1, \ldots, t_n).P \rfloor_A \cup \mathcal{P}; \mathcal{I}) \rightarrow_{\mathcal{T}} (\lfloor P_j \sigma_j \rfloor_{A_j} \cup \lfloor P \rfloor_A \cup \mathcal{P}; \mathcal{I}')$,
where $\sigma_j = mgu(t, t'_j)$ with $t = f(t_1, \ldots, t_n)$, and for $i \in \{1, \ldots, k\}$, if $(A, M_i) \in E$, then $I'_i = I_i \cup \{t\}$, else $I'_i = I_i$.

IN $\quad (\lfloor in(t').P \rfloor_A \cup \mathcal{P}; \mathcal{I}) \rightarrow_{\mathcal{T}} (\lfloor P\sigma \rfloor_A \cup \mathcal{P}; \mathcal{I}')$, if $(A, M_j) \in E, I_j \vdash t \,\&\, M_j \in \mathcal{M}$
where $\sigma = mgu(t, t')$, and if $(M_j, M_i) \in E$ $I'_i = I_i \cup \{t\}$, else $I'_i = I_i$.

IF-THEN $\quad (\lfloor \text{if } \Phi \text{ then } P \rfloor_A \cup \mathcal{P}; \mathcal{I}) \rightarrow_{\mathcal{T}} (\lfloor P \rfloor_A \cup \mathcal{P}; \mathcal{I}), \quad \text{if } [\![\Phi]\!]_G = 1.$

PAR $\quad (\lfloor P_1 | P_2 \rfloor_A \cup \mathcal{P}; \mathcal{I}) \rightarrow_{\mathcal{T}} (\lfloor P_1 \rfloor_A \cup \lfloor P_2 \rfloor_A \cup \mathcal{P}; \mathcal{I})$

REPL $\quad (\lfloor !P \rfloor_A \cup \mathcal{P}; \mathcal{I}) \rightarrow_{\mathcal{T}} (\lfloor P \rfloor_A \cup \lfloor !P \rfloor_A \cup \mathcal{P}; \mathcal{I})$

NEW $\quad (\lfloor new\ m.P \rfloor_A \cup \mathcal{P}; \mathcal{I}) \rightarrow_{\mathcal{T}} (\lfloor P\{m \mapsto m'\rfloor_A \cup \mathcal{P}; \mathcal{I})$,
where m' is a fresh name.

Figure 3: Transition system.

3.4 Security Property

We consider the route validity property. We say that a protocol satisfies route validity property and thus secure if it results in a *valid route*. In the follows, we define what is the valid route and what is the attack on a routing protocol.

Definition 1 (Valid route)**.** *Let $\mathcal{T} = (G, \mathcal{M}, S, D)$ be a topology with $G = (V, E)$, we say that a list $l = [A_1, \ldots, A_n]$ of agent names is an valid route in $\mathcal{T}$ if and only if for any $i \in \{1, \ldots, n-1\}$ $(A_i, A_{i+1}) \in E$ or $A_i, A_{i+1} \in \mathcal{M}$.*

We do not consider the case of wormhole attack where we have two successive non-neighbour malicious nodes.

After successfully executing a routing protocol, the source node stores the resulting received route. We assume that processes representing instances of routing protocols contain a process that has a special action of the form $out(end(l))$ which output the flag $end(l)$ at the end. The list l represent the established route so that we can check if the established route is valid. Checking whether a routing protocol ensures the validity of accepted route can be defined as a reachability property.

The attack on the configuration of a routing protocol can be modelled by the following definition.

Definition 2 (Attack on a configuration for a routing protocol)**.** *Let $\mathcal{T} = (G, \mathcal{M}, S, D)$ be a topology and K be a configuration. We say that K admits an* attack *in $\mathcal{T}$ if $K \rightarrow^*_{\mathcal{T}} (\lfloor out(end(l)).P \rfloor_A \cup \mathcal{P}; \mathcal{I})$ for some A, P, $\mathcal{P}$, $\mathcal{I}$, and some term l that is not an valid route in $\mathcal{T}$.*

The valid configuration for a routing protocol should satisfy this definition.

Definition 3 (Valid configuration)**.** *Let $\mathcal{T} = (G, \mathcal{M}, S, D)$ be a topology with $G = (V, E)$, and $\mathcal{I}$ be a set of sets representing the initial knowledge of the attackers. A configuration $K = (\mathcal{P}, \mathcal{I})$ is valid for the routing protocol $\mathcal{P}_{routing}$ and the routing role P_0 with respect to $\mathcal{T}$ if*

- *$\mathcal{P} = \lfloor P_0(S, D) \rfloor_S \uplus \mathcal{P}'$ and for every $\lfloor P' \rfloor_{A_1} \in \mathcal{P}'$ there exist $P(x_1, \ldots, x_n) \in \mathcal{P}_{routing}$, and $A_2, \ldots, A_n \in V$ such that $P' = P(A_1, \ldots, A_n)$.*
- *the only process containing a special action of the form $out(end(l))$ is $P_0(S, D)$ witnessing the storage of a route by the source node S.*

The first condition says that we only consider configurations that are made up using $P_0(S, D)$ and roles of the protocol, and the agent who executes the process is located at the right place. Moreover, we check whether the security property holds when the source and the destination are honest. The second condition ensures that the process witnessing the route is the process $P_0(S, D)$. Below we define the attack on a routing protocol $\mathcal{P}_{routing}$.

Definition 4 (Attack on $\mathcal{P}_{routing}$)**.** *We say that there is an attack on the routing protocol $\mathcal{P}_{routing}$ and the routing role P_0 given an initial knowledge $\mathcal{I}$ if there exist a topology $\mathcal{T} = (G, \mathcal{M}, S, D)$ and a configuration K that is valid for $\mathcal{P}_{routing}$ and P_0 with respect to $\mathcal{T}$, such that K admits an attack in $\mathcal{T}$.*

4 Reduction Procedure

We show that if there is an attack on route validity in a given topology then there is an attack in a smaller topology obtained by doing some reduction in the initial one. Our reduction procedure consists of two main steps:

1. Adding edges to the graph yielding a quasi-complete topology.
2. Merging nodes that have the same nature (honest or malicious) and same neighbours.

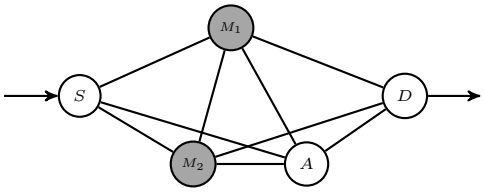
Finally in Section 4.3, we consider an arbitrary topology and apply our procedure on it. We end up with five particular topologies that contain at most three malicious nodes such that if there exists a network topology admitting an attack then there is an attack on one of these five topologies.

4.1 From an Arbitrary Topology to a Quasi-Complete One

Projecting nodes and reducing the size of the graph require that the nodes to be merged have the same nature and same neighbours. In order to ensure that most of the nodes have the same neighbours we first add edges to the graph. Actually, we add all edges except one. We show that the attack is preserved when we add these edges.

Definition 5 (Quasi-completion [6])**.** *Let $\mathcal{T} = (G, \mathcal{M}, S, D)$ be a topology with $G = (V, E)$, and A, B be two nodes in V that are not both malicious and such that $(A, B) \notin E$. The quasi-completion of $\mathcal{T}$ with respect to (A, B) is a topology $\mathcal{T}^+ = (G^+, \mathcal{M}, S, D)$ such that $G^+ = (V, E^+)$ with $E^+ = V \times V \setminus \{(A, B), (B, A)\}$.*

For example, a possible quasi-completion $\mathcal{T}_0^+$ of the topology $\mathcal{T}_0$ of Figure 2 is the one with respect to the pair (S, D) given below. As we see the graph is almost highly connected, the only missing edge is (S, D).



Definition 6 (Completion-friendly [6]). *A predicate p is completion-friendly if $[\![p(t_1,\ldots,t_n)]\!]_G = true$ implies that $[\![p(t_1,\ldots,t_n)]\!]_{G^+} = true$ for any ground terms $t_1,\ldots,t_n$ and any quasi-completion $\mathcal{T}^+ = (G^+,\mathcal{M},S,D)$ of $\mathcal{T} = (G,\mathcal{M},S,D)$. We say that a routing protocol (resp. a configuration) is completion-friendly if the predicates $\mathbf{P_D}$, i.e. the predicates that are dependent of the graph are completion-friendly.*

Predicates have to be completion-friendly so that their values are preserved when adding some edges to the graph.

Lemma 1 (Quasi-completion). *Let $\mathcal{T}$ be a topology, K_0 be a configuration that is completion-friendly. If there is an attack on K_0 in $\mathcal{T}$, then we can find two non-neighbour nodes $B,C \in V$ that are not both malicious and a topology $\mathcal{T}^+$ quasi-completion of $\mathcal{T}$ with respect to (B,C), such that there exists an attack on K_0 in $\mathcal{T}^+$.*

Proof. Let $\mathcal{T} = (G,\mathcal{M},S,D)$ be a topology with $G=(V,E)$ and K_0 be a configuration that is completion-friendly. Assume that there is an attack on K_0 in $\mathcal{T}$. Then, by the definition of the attack, there exist A, P, $\mathcal{P}$, $\mathcal{I}$ and $l_0 = [A_1,\ldots,A_n]$, such that $K_0 \to^*_{\mathcal{T}} (\lfloor out(end(l_0)).P \rfloor_A \cup \mathcal{P};\mathcal{I}) = K$ and l_0 is not an admissible path in $\mathcal{T}$, *i.e.* there exists $1 \le a \le n$ such that $(A_a,A_{a+1}) \notin E$ and ($A_a \notin \mathcal{M}$ or $A_{a+1} \notin \mathcal{M}$).

Let $\mathcal{T}^+ = (G^+,\mathcal{M},S,D)$ be a quasi-completion of $\mathcal{T}$ with respect to $(B,C) = (A_a,A_{a+1})$. We have, by definition of quasi-completion, that $G^+ = (V,E^+)$ with $E^+ = V\times V \setminus \{(A_a,A_{a+1}),(A_{a+1},A_a)\}$. We show by induction on the length r of a derivation $K_0 \to^r_{\mathcal{T}} K_r$ that K_r is completion-friendly and that $K_0 \to^r_{\mathcal{T}^+} K_r$. This will allow us to obtain that $K_0 \to^*_{\mathcal{T}^+} (\lfloor out(end(l_0)).P \rfloor_A \cup \mathcal{P};\mathcal{I})$, and as by definition of $\mathcal{T}^+$, l_0 is not an admissible path in $\mathcal{T}^+$ we conclude that K_0 admits an attack in $\mathcal{T}^+$.

For $r=0$, since K_0 is completion-friendly, we can easily conclude. Now, assume that $K_0 \to^{r-1}_{\mathcal{T}} K_{r-1}$ then, by induction hypothesis, we have that K_{r-1} is completion-friendly and $K_0 \to^{r-1}_{\mathcal{T}^+} K_{r-1}$. Since K_{r-1} is completion-friendly, then K_r is Completion-friendly as there is no rule that introduce new predicates or functions. We show that $K_{r-1} \to_{\mathcal{T}^+} K_r$ by case analysis on the rule involved in the step $K_{r-1} \to_{\mathcal{T}} K_r$:

Case of the rule IF-THEN: We have that $K_{r-1} = (\lfloor \text{if } \Phi \text{ then } P \rfloor_A \cup \mathcal{P};\mathcal{I})$, $K_r = (\lfloor P \rfloor_A \cup \mathcal{P};\mathcal{I})$ and $[\![\Phi]\!]_G = true$. Since K_{r-1} is completion-friendly and $[\![\Phi]\!]_G = true$ then $[\![\Phi]\!]_{G^+} = true$, it follows that we can apply the rule IF-THEN on K_{r-1} in $\mathcal{T}^+$, and thus we get that $K_{r-1} \to_{\mathcal{T}^+} K_r$.

Case of the rule IN: In such a case, we have that $K_{r-1} = (\lfloor in(t').P \rfloor_A \cup \mathcal{P};\mathcal{I})$, $K_r = (\lfloor P\sigma \rfloor_A \cup \mathcal{P};\mathcal{I}')$ where $\sigma = mgu(t,t')$, $(A,M_j) \in E$ for some $M_j \in \mathcal{M}$, $I_j \vdash t$ and for $i \in \{1,\ldots,k\}$, if $(M_j,M_i) \in E$ then $I'_i = I_i \cup \{t\}$, else $I'_i = I_i$.

We have that $E \subseteq E^+$, then $(A,M_j) \in E^+$ and if $(M_j,M_i) \in E$ then $(M_j,M_i) \in E^+$. Thus, we can easily conclude that $K_{r-1} \to_{\mathcal{T}^+} K_r$.

Rule COMM: We have that: $K_{r-1} = (\{\lfloor in(t'_j).P_j \rfloor_{A_j} \mid mgu(t,t'_j) \neq \bot, (A,A_j) \in E\} \cup \lfloor out(f(t_1,\ldots,t_n).P \rfloor_A \cup \mathcal{P};\mathcal{I})$ and $K_r = (\lfloor P_j\sigma_j \rfloor_{A_j} \cup \lfloor P \rfloor_A \cup \mathcal{P};\mathcal{I}')$ where $t = f(t_1,\ldots,t_n)$, $\sigma_j = mgu(t,t'_j)$, and for $i \in \{1,\ldots,k\}$, if $(A,M_i) \in E$ then $I'_i = I_i \cup \{t\}$, else $I'_i = I_i$.

As $E \subseteq E^+$, then $(A,A_j) \in E$ implies that $(A,A_j) \in E^+$, and $(A,M_i) \in E$ implies that $(A,M_i) \in E^+$. Thus, we have that $K_{r-1} \to_{\mathcal{T}^+} K_r$.

Case of the rules PAR, REPL, and NEW: These rules do not depend on the underlying graph. This allows us to easily conclude. □ □

4.2 Reducing the Size of the Topology

In this step, we merge nodes that have the same nature and same neighbours. The initial knowledge of malicious nodes are joined when they merged. In fact, sometimes one malicious node could do the job of several malicious nodes if we give it the required initial knowledge, for instance the case where we have a

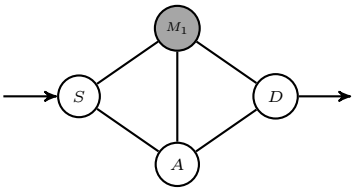
chain of malicious nodes. Also, in some cases existence or absence of some malicious nodes has no effect. We show that if there exists an attack in a given topology $\mathcal{T}$ then there exists an attack in a reduced topology $\rho(\mathcal{T})$ (some times written $\mathcal{T}\rho$) where ρ is a node renaming mapping.

Let $\mathcal{T} = (G, \mathcal{M}, S, D)$ be a topology with $G = (V, E)$ and E a reflexive and symmetric relation, and let ρ be a renaming on the agent names (not necessarily a one-to-one mapping). We say that the renaming $\rho : V \mapsto V$

- *preserves honesty* of $\mathcal{T}$ if $A, \rho(A) \in \mathcal{M}$ or $A, \rho(A) \notin \mathcal{M}$ for every $A \in V$.
- *preserves neighbourhood* of $\mathcal{T}$ if $\rho(A) = \rho(B)$ implies that $\{A' \in V \mid (A, A') \in E\} = \{B' \in V \mid (B, B') \in E\}$.

Given a term t, we denote by $t\rho$ the term obtained by applying the renaming ρ on t. This notation is extended to set of terms, configurations, graphs, and topologies. In particular, given a graph $G = (V, E)$, we denote $G\rho$ the graph $(V\rho, E')$ such that $E' = \{(\rho(A), \rho(B))|(A, B) \in E\}$. Note that when we apply a renaming ρ to a configuration $K = (\mathcal{P}, \mathcal{I})$ then the knowledge $I_i \in \mathcal{I}$ of $M_i \in \mathcal{M}$ is joined with the knowledge $I_{i'}$ of $M_i\rho = M_{i'}$ and the I_i is removed from $\mathcal{I}$.

Consider the quasi-completion $\mathcal{T}_0^+$ we seen before, a possible renaming ρ_0 that preserves neighbourhood and honesty and that allows us to reduce the size of the graph is defined by: $\rho_0(S) = S, \rho_0(A) = A, \rho_0(M_1) = \rho_0(M_2) = M_1, \rho_0(D) = D$. The resulting topology $\mathcal{T}_0^+\rho_0$ is given as follows:



Here, the two malicious nodes M_1 and M_2 are merged in M_1 then the knowledge I_2 corresponding to M_2 should be pooled with I_1 that of M_1. For instance, assume that we have initially $I_1 = \{M_1, S, D\}$, $I_2 = \{M_2, S, A\}$ and $\mathcal{I} = \{I_1, I_2\}$ then after merging we have that $I_1\rho_0 = \{M_1, S, D, A\}$ and $\mathcal{I}\rho_0 = \{I_1\rho_0\}$.

Note that ρ_0 does not preserve neighbourhood of the topology $\mathcal{T}_0$, this emphasises the importance of the completion step in order to make a safe merging.

Definition 7 (Projection-friendly [6])**.** *A predicate p is projection-friendly if $[\![p(t_1, \ldots, t_n)]\!]_G = true$ implies $[\![p(t_1\rho, \ldots, t_n\rho)]\!]_{G\rho} = true$ for any ground terms $t_1, \ldots, t_n$ and any renaming ρ that preserves neighbourhood and honesty. A function f over terms is projection-friendly if $f(t_1\rho, \ldots, t_n\rho)) = f(t_1, \ldots, t_n)\rho$ for any ground terms $t_1, \ldots, t_n$ and any renaming ρ that preserves neighbourhood and honesty. We say that a routing protocol (resp. a configuration) is projection-friendly if the predicates $\mathbf{P_I} \cup \mathbf{P_D}$ and the functions in $\mathcal{F}$ are projection-friendly.*

Lemma 2 (Reducing)**.** *Let $\mathcal{T}$ be a topology, K_0 be a configuration that is projection-friendly, and ρ be a renaming that preserves neighbourhood and honesty. If there is an attack on K_0 in $\mathcal{T}$, then there exists an attack on K_0' in $\mathcal{T}'$ where K_0' and $\mathcal{T}'$ are obtained by applying ρ on K_0 and $\mathcal{T}$ respectively.*

Proof. Let $\mathcal{T} = (G, \mathcal{M}, S, D)$ be a topology with $G = (V, E)$ and K_0 be a configuration that is projection-friendly. Assume that there is an attack on K_0 in $\mathcal{T}$. Then, by the definition of the attack, there exist A, P, $\mathcal{P}$, $\mathcal{I}$ and $l_0 = [A_1, \ldots, A_n]$, such that $K_0 \rightarrow_{\mathcal{T}}^* K = (\lfloor out(end(l_0)).P \rfloor_A \cup \mathcal{P}; \mathcal{I})$ and l_0 is not an admissible path in $\mathcal{T}$.

Let $K_0' = K_0\rho$ and $\mathcal{T}' = \mathcal{T}\rho = (G\rho, \mathcal{M}\rho, S\rho, D\rho)$ where $G\rho = (V\rho, E\rho)$. We show by induction on the length r of a derivation $K_0 \rightarrow_{\mathcal{T}}^r K_r$ that K_r is projection-friendly and $K_0' \rightarrow_{\mathcal{T}'}^r K_r'$ with $K_r' = K_r\rho$. This will allow us to obtain that $K_0' \rightarrow_{\mathcal{T}'}^* K'$ with $K' = K\rho$.

For $r = 0$, since $K_0' = K_0\rho$ and K_0 is projection-friendly, we can easily conclude. Assume that $K_0 \rightarrow_{\mathcal{T}}^{r-1} K_{r-1}$, then, by induction hypothesis, we have that K_{r-1} is projection-friendly and $K_0' \rightarrow_{\mathcal{T}'}^{r-1} K_{r-1}'$ with $K_{r-1}' = K_{r-1}\rho$. Since K_{r-1} is projection-friendly, then K_r is projection-friendly as there is no rule that introduce new predicates or functions. We show that $K_{r-1}' \rightarrow_{\mathcal{T}'} K_r'$ with $K_r' = K_r\rho$ by case analysis on the rule involved in the step $K_{r-1} \rightarrow_{\mathcal{T}} K_r$:

Case of the rule IF-THEN: We have that $K_{r-1} = (\lfloor \text{if } \Phi \text{ then } P \rfloor_A \cup \mathcal{P}; \mathcal{I})$, $K_r = (\lfloor P \rfloor_A \cup \mathcal{P}; \mathcal{I})$ and $[\![\Phi]\!]_G = true$. Since K_{r-1} is projection-friendly and $[\![\Phi]\!]_G = true$, then $[\![\Phi\rho]\!]_{G\rho} = true$, it follows that we can apply the rule IF-THEN on $K'_{r-1} = K_{r-1}\rho = (\lfloor \text{if } \Phi\rho \text{ then } P\rho \rfloor_{A\rho} \cup \mathcal{P}\rho; \mathcal{I}\rho)$, and thus we get that $K'_{r-1} \rightarrow_{\mathcal{T}'} K'_r$ with $K'_r = (\lfloor P\rho \rfloor_{A\rho} \cup \mathcal{P}\rho; \mathcal{I}\rho) = K_r\rho$.

Case of the rule IN: In such a case, we have that $K_{r-1} = (\lfloor in(t').P \rfloor_A \cup \mathcal{P}; \mathcal{I})$, $K_r = (\lfloor P\sigma \rfloor_A \cup \mathcal{P}; \mathcal{I}')$ where $\sigma = mgu(t, t')$, $(A, M_j) \in E$ for some $M_j \in \mathcal{M}$, $I_j \vdash t$, and for $i \in \{1, \ldots, k\}$, if $(M_j, M_i) \in E$ then $I'_i = I_i \cup \{t\}$, else $I'_i = I_i$. Furthermore, we have that $K'_{r-1} = K_{r-1}\rho = (\lfloor in(t'\rho).P\rho \rfloor_{A\rho} \cup \mathcal{P}\rho; \mathcal{I}\rho)$, $(A\rho, M_j\rho) \in E\rho$ since $(A, M_j) \in E$, $M_j\rho \in \mathcal{M}\rho$ since $M_j \in \mathcal{M}$ and ρ preserve honesty, $I_i\rho \vdash t\rho$, and if $(M_j, M_i) \in E$ for some $M_i \in \mathcal{M}$, then $(M_j\rho, M_i\rho) \in E\rho$ and $M_i\rho \in \mathcal{M}\rho$ since ρ preserves neighbourhood and honesty. Thus, $K'_{r-1} \rightarrow_{\mathcal{T}'} (\lfloor (P\rho)\sigma' \rfloor_{A\rho} \cup \mathcal{P}\rho; \mathcal{I}'\rho) = K'_r$, where $\sigma' = mgu(t\rho, t'\rho)$. Note that, $(P\rho)\sigma' = (P\sigma)\rho$, and thus $K'_r = (\lfloor (P\sigma)\rho \rfloor_{A\rho} \cup \mathcal{P}\rho; \mathcal{I}'\rho) = K_r\rho$.

Case of the rule COMM: In this case $K_{r-1} = (\{\lfloor in(t'_j).P_j \rfloor_{A_j} \mid mgu(t, t'_j) \neq \bot, (A, A_j) \in E\} \cup \lfloor out(f(t_1, \ldots, t_n).P \rfloor_A \cup \mathcal{P}; \mathcal{I})$ and $K_r = (\lfloor P_j\sigma_j \rfloor_{A_j} \cup \lfloor P \rfloor_A \cup \mathcal{P}; \mathcal{I}')$ where $t = f(t_1, \ldots, t_n)$, $\sigma_j = mgu(t, t'_j)$, and for $i \in \{1, \ldots, k\}$, if $(A, M_i) \in E$ then $I'_i = I_i \cup \{t\}$, else $I'_i = I_i$.

We have that, $K'_{r-1} = K_{r-1}\rho = (\{\lfloor in(t'_j\rho).P_j\rho \rfloor_{A_j\rho} \mid mgu(t\rho, t'_j\rho) \neq \bot, (A\rho, A_j\rho) \in E'\} \cup \lfloor out(f(t_1, \ldots, t_n)\rho).P\rho \rfloor_{A\rho} \cup \mathcal{P}\rho; \mathcal{I}\rho)$, $f(t_1, \ldots, t_n)\rho = f(t_1\rho, \ldots, t_n\rho)$ since f is projection-friendly, and $(A\rho, M_i\rho) \in E'$ if $(A, M_i) \in E$, then $K'_{r-1} \rightarrow_{\mathcal{T}'} (\lfloor (P_j\rho)\sigma'_j \rfloor_{A_j\rho} \cup \lfloor P\rho \rfloor_{A\rho} \cup \mathcal{P}\rho; \mathcal{I}'\rho) = K'_r$, $\sigma'_j = mgu(t\rho, t'_j\rho)$. Thus, as $(P_j\rho)\sigma'_j = (P_j\sigma_j)\rho$, $K'_r = (\lfloor (P_j\sigma_j)\rho \rfloor_{A_j\rho} \cup \lfloor P\rho \rfloor_{A\rho} \cup \mathcal{P}\rho; \mathcal{I}'\rho) = K_r\rho$.

Case of the rules PAR, REPL, and NEW: These rules do not depend on the underlying graph. This allows us to easily conclude.

Hence, we have that $K'_0 \rightarrow^*_{\mathcal{T}'} (\lfloor out(end(l_0\rho)).P\rho \rfloor_{A\rho} \cup \mathcal{P}\rho; \mathcal{I}\rho) = K' = K\rho$. In order to conclude that there is an attack on K'_0 in $\mathcal{T}'$, it remains to show that $l_0\rho = [A_1\rho, \ldots, A_n\rho] = [A'_1, \ldots, A'_n]$ is not an admissible path in $\mathcal{T}'$. First, we want to note that: (i) If $B \notin \mathcal{M}$ then $B\rho \notin \mathcal{M}\rho$. Assume that, $B \notin \mathcal{M}$ then $B\rho \in \mathcal{M}\rho$ then there exists $C \in \mathcal{M}$ such that $B\rho = C\rho$. Thus, as ρ preserve honesty we have that B and C are both malicious or both honest, which leads to a contradiction. (ii)If $(B_1, B_2) \notin E$ then $(B_1\rho, B_2\rho) \notin E\rho$. Let $B_1, B_2 \in V$, $\rho(B_1) = D_1$ and $\rho(B_2) = D_2$ such that$(B_1, B_2) \notin E$. Suppose that $(D_1, D_2) = (B_1\rho, B_2\rho) \in E\rho$, then by definition of $E\rho$ there exist two nodes $C_1, C_2 \in V$ such that $\rho(C_1) = D_1$, $\rho(C_2) = D_2$ and $(C_1, C_2) \in E$. Since $\rho(B_1) = \rho(C_1) = D_1$ and ρ preserves neighbourhood, we get that $N_G(B_1) = N_G(C_1)$. It follows that B_1 and C_2 are neighbours in G since C_1 and C_2 are neighbours in G. The same we have that $N_G(B_2) = N_G(C_2)$. Thus B_1 and B_2 are also neighbours in G, *i.e.* $(B_1, B_2) \in E$ which leads to a contradiction. Hence, $(B_1\rho, B_2\rho) \notin E\rho$.

Now, as l_0 is not an admissible path in $\mathcal{T}$, there exists $1 \leq a \leq n$ such that $(A_a, A_{a+1}) \notin E$ and ($A_a \notin \mathcal{M}$ or $A_{a+1} \notin \mathcal{M}$). Then, $(A_a\rho, A_{a+1}\rho) \notin E\rho$ and ($A_a\rho \notin \mathcal{M}\rho$ or $A_{a+1}\rho \notin \mathcal{M}\rho$). Hence, $l_0\rho$ is not an admissible path in $\mathcal{T}'$ and we can conclude. □ □

4.3 Five Topologies are Sufficient

We show that for a protocol $\mathcal{P}_{routing}$ there is an attack on an arbitrary topology if and only if there is an attack on one of five particular topologies. Our result holds for an unbounded number of sessions since we consider arbitrarily many instances of the roles occurring in $\mathcal{P}_{routing}$.

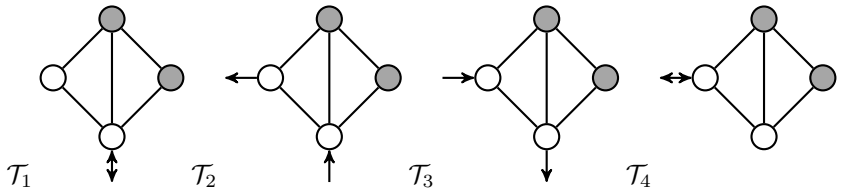
Theorem 1 (Five topologies)**.** *Let $\mathcal{P}_{routing}$ be a routing protocol and P_0 be a routing role which are both completion-friendly and projection-friendly and $\mathcal{I}$ be a set of knowledge. There is an attack on $\mathcal{P}_{routing}$ and P_0 given the knowledge $\mathcal{I}$ for some $\mathcal{T}$, if and only if, there is an attack on $\mathcal{P}_{routing}$ and P_0 given the knowledge $\mathcal{I}$ for one of five particular topologies $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4$ and $\mathcal{T}_5$.*

Proof. If there is an attack on $\mathcal{P}_{routing}$ and P_0 given $\mathcal{I}$ for one of the five particular topologies, we easily conclude that there is an attack on $\mathcal{P}_{routing}$ and P_0 given $\mathcal{I}$ for some topology $\mathcal{T}$. We consider now the other implication. Let $\mathcal{T} = (G, \mathcal{M}, S, D)$ be a topology with $G = (V, E)$, $\mathcal{I}$ be a set of knowledge and $K = (\mathcal{P}, \mathcal{I})$ be a valid configuration for $\mathcal{P}_{routing}$ and P_0 with respect to $\mathcal{T}$, such that there is an attack on K in $\mathcal{T}$. Without lost of generality, we assume that V contains at least three distinct honest nodes and three distinct malicious nodes. Note that otherwise, it is easy to add some nodes in the topology $\mathcal{T}$ and still preserving the existence of an attack.

First, it is easy to see that K is completion-friendly as $\mathcal{P}_{routing}$ and P_0 are both completion-friendly. Thanks to the Lemma 1, we deduce that there exists two non-neighbour nodes $B, C \in V$ that are not both malicious and a topology $\mathcal{T}^+ = (G, \mathcal{M}, S, D)$, a quasi-completion of $\mathcal{T}$ with respect to (B, C), such that there is an attack on K in $\mathcal{T}^+$. As $\mathcal{T}^+$ is a quasi-completion of $\mathcal{T}$ with respect to a pair (B, C), then the neighbours of B in G^+ denoted $N_{G^+}(B) = V \setminus \{C\}$, $N_{G^+}(C) = V \setminus \{B\}$, and $N_{G^+}(W) = V$ for any $W \in V \setminus \{B, C\}$. Since we have assumed that V contains at least three distinct nodes that are not in $\mathcal{M}$ and three distinct nodes in $\mathcal{M}$, we deduce that $V \setminus \{B, C\}$ contains at least an honest node let us say A and a malicious one let us say M. Let ρ be a renaming on the agent names such that for any $W \in V \setminus \{B, C\}$, $\rho(W) = A$ if $W \notin \mathcal{M}$ and $\rho(W) = M$ else. Clearly, we have that ρ preserves honesty and neighbourhood. Thanks to Lemma 2, we deduce that there is an attack on $K' = K\rho$ in $\mathcal{T}' = (G\rho, \mathcal{M}\rho, S\rho, D\rho) = \mathcal{T}^+\rho$.

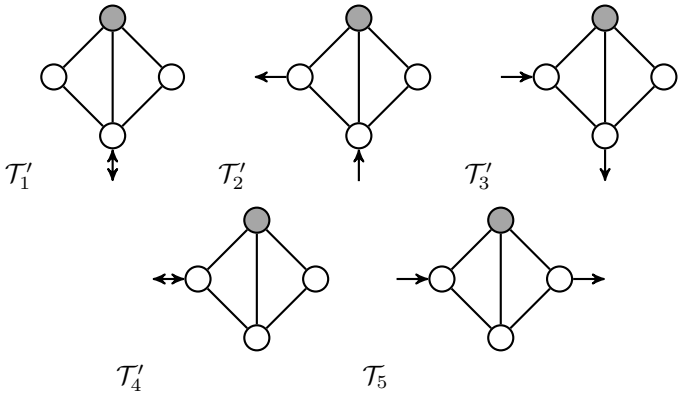
The topology $\mathcal{T}'$ has four nodes: one honest, one malicious and two nodes B, C. We distinguish cases depending in the nature of the nodes B and C:

1. B honest and C malicious (the reverse is the same due to symmetry). In this case $\mathcal{T}'$ has two honest nodes, thus according to the position of the source and destination we have the following four possibilities:



   Note that the topology $\mathcal{T}_4$ can obtained only if the source and destination are the same in the original topology.

2. Both are honest. So $\mathcal{T}'$ has three honest nodes in this case. Depending on the position of the source and destination we have nine possibilities, but due to symmetry four of them can be eliminated. This results in only five topologies:



   Again the topologies $\mathcal{T}'_1, \mathcal{T}'_2, \mathcal{T}'_3$ and $\mathcal{T}'_4$ are subsumed by $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ and $\mathcal{T}_4$ respectively, since if there is an attack in $\mathcal{T}'_i$ for $i \in \{1, 2, 3, 4\}$, then this attack can be mounted in $\mathcal{T}_i$ where an honest node is now malicious.

So $\mathcal{T}'$ is one of the five topologies $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4$ and $\mathcal{T}_5$. Now, since $\mathcal{P}_{routing}$ and P_0 do not contain any names, Definition 3 is satisfied and thus $K' = (\mathcal{P}\rho, \mathcal{I}\rho)$ is a valid configuration with respect to $\mathcal{T}'$. □ □

5 Comparison Between the Two Models

We show the equivalence of cooperative model and non-cooperative one when considering all possible topologies. First, if we have a topology $\mathcal{T}$ that a protocol admits an attack under the cooperative model,

we show how to obtain a topology $\mathcal{T}'$ from such $\mathcal{T}$ that this protocol admits an attack on it under the non-cooperative model. Then, we present the equivalence theorem.

Lemma 3 (Preservation of the attack)**.** *Let K be a configuration that is completion friendly. If there exists a topology $\mathcal{T}$ such that the configuration K admits an attack in $\mathcal{T}$ under the cooperative model then there exists a topology $\mathcal{T}'$ to be obtained from $\mathcal{T}$ such that K admits an attack in $\mathcal{T}'$ under the non-cooperative model.*

Proof. Let $\mathcal{T} = (G, \mathcal{M}, S, D)$ be a topology with $G = (V, E)$ and K be a configuration that is completion friendly. Suppose that there is an attack on K in $\mathcal{T}$ then, by the definition of the attack, there exist A, P, $\mathcal{P}$, $\mathcal{I}$ and $l_0 = [A_1, \ldots, A_n]$ such that $K \rightarrow^*_{\mathcal{T}} (\lfloor out(end(l_0)).P \rfloor_A \cup \mathcal{P}; \mathcal{I})$ and l_0 is not an valid route in $\mathcal{T}$.

Let $\mathcal{T}' = (G', \mathcal{M}, S, D)$ be a topology such that $G' = (V, E')$ with $E' = E \cup (\mathcal{M} \times \mathcal{M})$. Since l_0 is inadmissible in $\mathcal{T}$, it is also inadmissible in $\mathcal{T}'$ according to the definition. To deduce that there is an attack on K in $\mathcal{T}'$ we show that $K \rightarrow^*_{\mathcal{T}'} (\lfloor out(end(l_0)).P \rfloor_A \cup \mathcal{P}; \mathcal{I})$. Note that we assume the same initial knowledge for all attackers, if its not the case we can reach this state by applying successively the COMM rule a certain number of times as all malicious nodes are connected in $\mathcal{T}'$. For each rule involved in the transition $K_r \rightarrow_{\mathcal{T}} K_{r+1}$ under the cooperative model we show the equivalence rule or rules in $\mathcal{T}'$ under non-cooperative model to have $K_r \rightarrow^*_{\mathcal{T}'} K_{r+1}$

- **Case of the rule IF-THEN:** Since K is completion friendly then any formula ϕ that is true for $\mathcal{T}$, its true also for $\mathcal{T}'$. Thus, the rule IF-THEN can also be applied in $\mathcal{T}'$ in this case.
- **Case of the rule IN:** Since $E \subseteq E'$ the same rule can be applied and as all malicious nodes are connected in $\mathcal{T}'$ the sent message is received by all attackers so the knowledge remains equal.
- **Case of the COMM:** Since $E \subseteq E'$ we can apply the rule COMM, but in case where we have a malicious node neighbour of the node that plays the role of sender the rule COMM should be followed by a certain number (equal to the number of other malicious nodes) of rule IN application. The last step is to share in an indirect way the message received by one of the malicious nodes as it is a neighbour of the sender.
- **Case of the rules PAR, REPL, and NEW:** These rules do not depend on the underlying graph. So same rules can be applied in $\mathcal{T}'$. □

□

Theorem 2 (Equivalence)**.** *Let $\mathcal{P}_{routing}$ be a routing protocol and P_0 be a routing role and $\mathcal{I}$ be a set of knowledge. We have that $\mathcal{P}_{routing}$ and P_0 given the knowledge $\mathcal{I}$ are secure for any $\mathcal{T}$ in the cooperative model, if and only if, $\mathcal{P}_{routing}$ and P_0 given the knowledge $\mathcal{I}$ are secure for any $\mathcal{T}$ in the non-cooperative model.*

Proof. First direction: in non-cooperative model the malicious nodes have weaker abilities. So, if there is no attack on $\mathcal{P}_{routing}$ and P_0 given $\mathcal{I}$ in cooperative model then there is no attack on $\mathcal{P}_{routing}$ and P_0 in the non-cooperative model.

Second direction: Suppose that there is no attack on $\mathcal{P}_{routing}$ and P_0 given $\mathcal{I}$ for any topology in the non-cooperative model. Assume that there exists a topology $\mathcal{T}$ such that there is an attack on $\mathcal{P}_{routing}$ and P_0 given $\mathcal{I}$ in $\mathcal{T}$ for cooperative model, then by the definition of the attack there are a configuration K that is valid for $\mathcal{P}_{routing}$ and P_0 such that there is an attack on K in $\mathcal{T}$ under cooperative model. Then, by Lemma 3, there is a topology $\mathcal{T}'$ obtained from $\mathcal{T}$ such that there is an attack in it on the configuration K for the non-cooperative model and thus an attack on $\mathcal{P}_{routing}$ and P_0 given $\mathcal{I}$ in $\mathcal{T}'$ which leads to a contradiction. □ □

Considering one fix topology $\mathcal{T}$ this equivalence do not hold anymore as we can find an attack on a protocol in $\mathcal{T}$ under the cooperative model, while this protocol is secure in $\mathcal{T}$ under the non-cooperative model. This due to the fact that under the cooperative model we give the malicious nodes a powerful capabilities that are not exists in reality, this leads to a false positive attacks that can not be mounted in practice. Having a fixed known topology one could prefer to verify the used protocol under the non-cooperative model which gives a more realistic security level.

6 Conclusion

We consider the non-cooperative attacker model where there are multiple attackers working independently, so that no one share any of its knowledge with the others. We give a reduction proof: when looking for attacks on route validity in presence of multiple independent attackers if there is an attack in a certain topology then there is an attack in a smaller one. Then, we show that there is an attack on an arbitrary topology if and only if there is an attack on one of five particular topologies, each of them having only four nodes. This result facilities verification of routing protocols as we have to check only five small topologies. Finally, we show that a protocol is secure in any topology under the cooperative model if and only if it is secure for any topology under the non-cooperative model.

For future work, it could be interesting to develop a tool that able to solve multiple attackers constraints, so that we can reason on the five topologies one by one in order to verify ad-hoc network routing protocols.

References

[1] T. R. Andel, G. Back, and A. Yasinsac. Automating the security analysis process of secure ad hoc routing protocols. *Simulation Modelling Practice and Theory*, 19(9):2032–2049, 2011. 1

[2] M. Arnaud, V. Cortier, and S. Delaune. Modeling and verifying ad hoc routing protocols. In *CSF*, pages 59–74. IEEE Computer Society, 2010. 1, 3.3

[3] T. Avanesov, Y. Chevalier, M. Rusinowitch, and M. Turuani. Satisfiability of general intruder constraints with and without a set constructor. *CoRR*, abs/1103.0220, 2011. 1

[4] L. Buttyn and I. Vajda. Towards provable security for ad hoc routing protocols. In *In Proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN*, pages 94–105. ACM Press, 2004. 1

[5] Y. chun Hu, A. Perrig, and D. B. Johnson. Wormhole attacks in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24:370–380, 2006. 1

[6] V. Cortier, J. Degrieck, and S. Delaune. Analysing routing protocols: Four nodes topologies are sufficient. In P. Degano and J. D. Guttman, editors, *POST*, volume 7215 of *Lecture Notes in Computer Science*, pages 30–50. Springer, 2012. 1, 3.1, 3.3, 5, 6, 7

[7] D. Dolev and A. C.-C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983. 1, 3.2

[8] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, 11(1-2):21–38, 2005. 1

[9] D. B. Johnson, D. A. Maltz, and J. Broch. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, pages 139–172. Addison-Wesley, 2001. 1

[10] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. W. Chang. Preventing wormhole attacks on wireless ad hoc networks: A graph theoretic approach. In *in IEEE Wireless Communications and Networking Conference WCNC*, pages 1193–1199, 2005. 1

[11] L. Mazaré. Satisfiability of dolev-yao constraints. *Electr. Notes Theor. Comput. Sci.*, 125(1):109–124, 2005. 1

[12] S. Nanz and C. Hankin. A framework for security analysis of mobile wireless networks. *Theoretical Computer Science*, 367:2006, 2006. 1, 3.1

[13] P. Papadimitratos and Z. Haas. Secure Routing for Mobile Ad hoc Networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 2002. 1, 2.1

[14] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. A secure routing protocol for ad hoc networks. In *ICNP*, pages 78–89. IEEE Computer Society, 2002. 1