SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2015; **8**:418–430
Published online 4 July 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.990

RESEARCH ARTICLE

PoliCon: a policy conciliation framework for heterogeneous mobile ad hoc networks

Soumya Maity[1], Soumya K. Ghosh[1]* and Ehab Al-Shaer[2]

[1] School of Information Technology, Indian Institute of Technology Kharagpur, India 721302
[2] Department of Software and Information Systems, University of North Carolina, Charlotte, NC 28223, USA

ABSTRACT

It is increasingly important to implement a conflict-free access control policies for co-allied networks where different organizations are involve for a common goal. Mobile ad hoc networks are widely used for mission critical situations where teams from different organizational networks cooperate to form a single network to implement their respective operations. These teams (or quads) have different sets of local policies enforced for their own security resulting heterogeneity in access control. Each team wants to preserve its access control policies at a maximum level. Moreover, a set of *allied policies* govern the cooperation and interaction between the different teams, which may conflict with their local policies. The policy conflicts arise from the transitivity of policy rules, mobility of the nodes, cooperative behaviors, and so on. In addition, the policy rules may be temporal or static. To achieve the successful completion of the mission, it may be required to compromise with the stringency of the enforcement of the conflicting rules for the quads. In this paper, we propose an automated and formal framework to find the optimal conciliation of the policy rules to preserve the mission and thus ensure minimal compromise with the enforcement of policy for each quad. The efficacy of the work lies on optimizing the enforcement of access control policies to achieve the coalition instead of negating the policy. Copyright © 2014 John Wiley & Sons, Ltd.

KEYWORDS
policy conciliation; MANET; access control

***Correspondence**
Soumya K. Ghosh, School of Information Technology, Indian Institute of Technology, Kharagpur, India 721302.
E-mail: skgkgp@iitkgp.ac.in

1. INTRODUCTION

Policy conciliation refers to resolution of policy conflicts between two or more allied parties by mutually compromising some profit to achieve a common goal. In this paper, the conflicts between policies of different networks have been resolved by relaxing the strictness level of the policy implementation of the corresponding organization. The co-allied networks or teams are referred as *quads*. Each quad has its own access control policies according to the different types of security requirements. Each node follows a set of security protocols to implement the specifications. So, on the basis of local policy, the network is *heterogeneous*. However, the collection or summation of these distributed implementations may violate the overall security. There may exist hidden access paths for the transitive property of the access policies. Moreover, the dynamic topology, changing values of trust relation, and absence of network perimeter increase the chance of security violations. As an example, a given topology may ensure the implementation to be conforming the specification, but a certain change in topology due to the mobility of the nodes may create a violation.

In this paper, a formal mechanism to find the optimized set of trust-based policy implementation for consolidated policies has been proposed. As formal verification is a promising approach to correctly and completely verify implementation with specification, the problem has been modeled to a satisfiability problem. The efficiency of this approach has been discussed later. The model of *mobile ad hoc network* (MANET) has been taken into consideration as it is a widely used in mission critical organizations for communication. Though, the same framework can be extended for any networks. The framework named policy conciliation (*PoliCon*), supported by the formal mechanism, is implemented. This framework is also capable to detect any local policy conflicts within a quad. The conciliation of the policy implementation will be correct,

complete, and sound. Correctness means there does not exist any valid policy for which the allied policies are conflicting. The completeness signifies that no other policy rule is required to achieve the allied access control requirement. Soundness ensures each of the trust and policy models is valid (decision based on a set of valid evidences). Suggestions for maximal level implementation of the conflicting policy to the allied parties will be provided by PoliCon framework. These policies are usually trust-based and can be temporal as well as static.

Enforcement of the policy rules set by the administrator is a different research issue and beyond the scope of this paper. In this paper, we have studied the effect of conflicting policies for co-allied MANET and proposed an optimized conciliation of the policy. It has been assumed that the network is capable to enforce the policy given by the network administrator over the distributed nodes. Pervasive use of MANET challenged the researchers in various security related issues. Many research attempts have been made on enforcing policy-based security for MANET to ensure access control. Among which work by Luo *et al*. [1], Alicherry and Keromytis [2], Mulert *et al*. [3], and so on can be mentioned. For semi-infrastructured MANETs, our earlier work [4] proposed a mechanism for implementing access control. In a recent work [5], we have developed a framework to enforce policy-based access control in MANET. Proposing a new trust model is also not in the scope of this paper. We assume an underlying trust model gives a trust value of all other nodes with respect to a given nodes. The range of trust value is scaled in 0 to 1. We refer to our earlier work [6] for details of the trust model.

In this paper, we have formally modeled the network and the policies (both local and allied). We have modeled the implementation of each policy in a varying range of strictness of enforcement. The highest level of implementation is keeping the rule as it is without compromising, and the lowest level of implementation is removing the policy rule from the quad. The implementation of the policy varies in this range in the suggested conciliation. The framework finds the solution with the highest level of enforcement of each of the policy rule that does not conflict with allied policy.

1.1. Motivating example

Mobile ad hoc network is actively used in mission critical scenario where people from different organizations with different domain of expertise, work together for a rescue mission. In that scenario, each of them requires different set of policies for their operation. Reachability and avoidance of disconnectivity are critical issues.

We have taken an example of co-allied MANET where coalition has been formed between three organizations, namely, Quad1, Quad2, and Quad3. For simplicity of this example, we assume that the only service running in each note is *getLocation()*, which gives the present coordinate of the requested node.

Each of the nodes can request another node to share its current location, and the requested node sends the answer back with its own location. There is a policy enforced on all the nodes of Quad1: "`Share the location information with nodes which are only from the same quad if there is a trusted route.`" Now, the coalition between Quad1, Quad2, and Quad3 requires an allied policy, "`Nodes of Quad2 needs to know the area where a node from Quad1 is located.`" Logically, the former local policy of Quad1 is conflicting the allied policy. So, Quad1 needs to compromise with the policy. Just removing or negating the local policy from Quad1 is not a desirable solution. Moreover, transitivity of policy rules, mobility of the nodes, and variation of trust values over time lead to conflicts in policies. Our framework checks for such conflicts and suggests the optimal conciliation. So, as a better solution, the proposed PoliCon framework suggests an optimized *conciliated policy* as,

"`Share the exact location information nodes from Quad1 if there is a trusted route`

or

`Share approximate location (like a defined boundary) information with Quad2 if there is very high trust value of the requesting node.`"

Instead of negating the policy itself, the proposed *PoliCon* framework actually negotiates with the implementation level of the policy.

The rest of the paper is organized as follows. Section 2 describes the overview of PoliCon framework. The framework has three units, and each of the units is elaborately explained. The formal modeling of network instance and network access policies is explained in Sections 3 and 4, respectively. Section 5 represents the process of optimization of conciliated policies. The implementation and analysis on the results are shown in Section 6. Section 8 concludes the paper with remarks and future direction of the work.

2. OVERVIEW OF POLICON FRAMEWORK

Satisfiability (SAT) based formal framework for generating conciliation of policy rules between the quads (allied parties) has been proposed in this work. The input to this framework is the different parameters of the network at particular instance of time. The output is an optimal conciliated policy implementation. It has three modules (refer to Figure 1), namely, (i) *network instance modeler* (NIM), (ii) *network access modeler*, and (iii) *conciliation optimizer*.

The *NIM* parses the specification and generates a formal model of the network instance, which represents the network topology. As the nodes are mobile, the topology

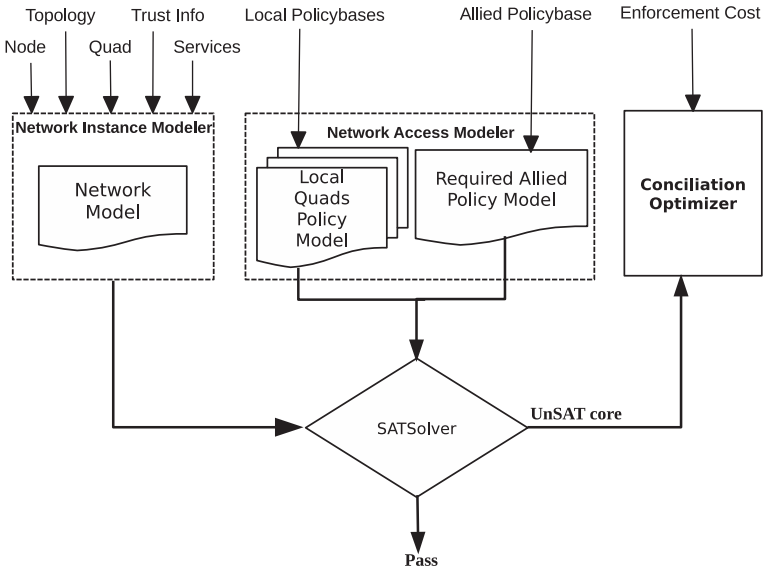


Figure 1. Overview of PoliCon framework.

is constantly changing. So, taking a snapshot at a particular instance is not enough. Hence, the network model should capture the mobility. We have used a mobility model to generate a valid set of the topologies that the network follows. We call this set as *closure of topology*. The topology comprises of various network resources. The network resources and their properties have been encoded into Boolean logic. In this work, we are not proposing a completely new mobility model. Rather, we assume that the mobility of the nodes are given as semantics [7]. For simulation, random way-point model [8] has been considered. For the purpose of verification using Boolean SAT based decision procedure, our framework converts the network instance model into conjugal normal form (CNF) clauses [9].

The local policy specifications are encoded into Boolean logic by the *network access modeler*. The allied policy requirement is also modeled under the same formalism and converted into CNF clauses. Finally, our framework uses zChaff SAT solver for checking the SAT of the conjunction of these CNF clauses. The satisfiable result (i.e., does not find any counter-example) indicates that there is no violation of policies or the policies are nonconflicting and consolidatable. Otherwise, the conflicting policies from unsatisfiable core are extracted. Each of the possible implementation level of the conflicting policies is assigned with a numeric weight. Finally, these implementation levels along with the allied policy are modeled as a minimum-cost SAT problem [10,11] for generating optimized PoliCon. The satisfiable instance of the *Min-Cost SAT* represents the conflict-free conciliated policy implementation with minimal cost. PoliCon framework systematically generates the optimal solution by iterating the *Min-Cost SAT* procedure for all the conflicting rules.

3. NETWORK INSTANCE MODEL

Network instance modeler of the proposed PoliCon framework takes the responsibility of converting the network instance into Boolean clauses. For this purpose, we have modeled the network instance.

A MANET is a network formed by n nodes (n being a real positive number). N denotes the set nodes. Every node has a unique ID ($N_{id} \in N$) and a Quad ($q \in Q$). The unique ID is an address within a fixed global address space. The quad is one the set of allied parties (Q). We assume a random distribution of the node IDs. This distribution is ensured by mapping some device specific address, for example, Internet host names, IP addresses, or MAC addresses, to the node IDs using a strong hash function (not further specified here). Consider the resulting addresses as given by a static mapping from nodes to some abstract domain of addresses. The underlying network protocols ensures that node $N_i \in N$ can send data packet to another node $N_j \in N, i \neq j$ if N_j is reachable to N_i.

Definition 1 (Network instance). *A network instance (NI) is defined as two tuples $\langle N, I \rangle$, where N is the set of finite hosts, and I is the connectivity matrix of all the nodes in N.*

A node $n \in N$ has two attributes, ID and Quad. n.ID is a positive real number i such that $\forall n' \in \{N - n\} \mid n'.ID \neq i$. In the paper, n_i conventionally represents $n_i \in N \wedge N_i.ID = i$. Q is the set of the identifiers of the allied parties. n.Quad must be in Q. Connectivity matrix I is a $|N| \times |N|$ matrix where the nondiagonal entries $a_{ij} = 1$ if n_j is in the radio range of n_i, eventually n_i can directly communicate with n_j, else $a_{ij} = 0$. If $a_{ij} = 0$ in I, then n_j is said to be a neighbor of n_i, denoted as Neighbor(n_i, n_j).

DOI: 10.1002/sec

A node $n \in N$ has some properties such as mobility, trust, and reachability, and it can execute some functions such as request, response, forward, discard, and attack. We assumed our model to be closed under these properties and functions.

Property (Mobility)**.** *Each node $n \in N$ has a coordinate identifying its geographic position on the plane. As nodes may move, their positions change over time. The current position of a node is always known. The position of n is defined by POS(n). Mobility of a node n after time interval t, denoted as $\mu(n,t)$, is the property of the node n for which the value of POS(n) at $t = 0$ may differ from the value of POS(n) at $t = t$. As the position of the nodes changes, the connectivity matrix also changes. So, if at time instance $t = 0$, the connectivity matrix was $I(0)$ and at time = t, because of the mobility of the nodes, connectivity matrix becomes $I(t)$. $I(0) \rightsquigarrow_t I(t)$ denotes the connectivity matrix $I(0)$ changed to $I(t)$ after time interval t. Mobility of a network instance $\Upsilon_{NI}(N,I,t)$ is defined as $(\forall n \in N, \sum \mu(n,t) \Rightarrow I(0) \rightsquigarrow_t I(t)) \Rightarrow NI'(N,I(t),t)$. All the possible values of $I(t)$ comprises to form a set of connectivity matrices called closure of topology (I*).*

We model the impact of mobility, deletions, and additions of nodes by representing the network topology at any point in time with an NN adjacency matrix denoted by $I(t)$. The matrix is as follows:

$$I(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) & a_{13}(t) & \dots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & a_{23}(t) & \dots & a_{2n}(t) \\ a_{31}(t) & a_{32}(t) & a_{33}(t) & \dots & a_{3n}(t) \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ a_{n1}(t) & a_{n2}(t) & a_{n3}(t) & \dots & a_{nn}(t) \end{bmatrix}$$

where,

$a_{ij}(t) = 1$, if nodes i and j are directly connected at time t
$a_{ij}(t) = 0$, otherwise.

a^*_{ij} is defined as, $\bigvee_{t=0}^{t} a_{ij}(t)$, which means, if at any point of time, nodes i and j are directly connected, $a^*_{ij} = 1$, otherwise 0.

Closure of topology I^* is defined by the matrix,

$$I^* = \begin{bmatrix} a^*_{11} & a^*_{12} & a^*_{13} & \dots & a^*_{1n} \\ a^*_{21} & a^*_{22} & a^*_{23} & \dots & a^*_{2n} \\ a^*_{31} & a^*_{32} & a^*_{33} & \dots & a^*_{3n} \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ a^*_{n1} & a^*_{n2} & a^*_{n3} & \dots & a^*_{nn} \end{bmatrix}$$

This matrix captures the topology of the network over time interval t. Hence, this matrix is used for modeling the network instead of a snapshot of the topology at a particular instance.

Property (Trust)**.** *All the nodes in the network might not be equally trustworthy. A proper trust model based on recommendation, behavioral analysis, central certification, and so on assigns a trust value to a node with respect to a particular node. A number of trust models* [12] *have been proposed by the researcher, which are able to quantitatively calculate trust value of other nodes over time. Proposing a new trust model is not in the scope of this paper. We assume that an underlying trust model gives a trust value of all other nodes with respect to a given nodes. The range of trust value is scaled in 0 to 1. The detail trust model has been described in our previous work* [6]. *The underlying trust model ensures that each node N_i measures trust value $\tau_(n_i,n_j)$ of another node N_j on the basis of a set of evidences E_i. These evidences E_i are captured by individual nodes N_i. On the basis of the set of trust values $(T = \tau_{n_i}(n_j), \forall i \forall j, i \neq j)$, the node performs certain actions or takes certain decisions, denoted as A_i. Trust $\tau(n_i,n_j)$ of a node n_j with respect to node N_i is defined as a real number within range [0,1]. $\tau(n_i,n_j) = 0$ denotes n_j is not trustworthy to n_i. The untrustworthiness $u(n_i,n_j)$ of n_j with respect to n_i is defined as $(1 - \tau(n_i,n_j))$.*

Property (Reachability)**.** *A node $n_i \in N$ is said to be reachable to $n_j \in N$ if n_i can communicate with n_j, directly or indirectly. It is denoted as $Reachable(n_i,n_j)$. The necessary condition for every node must hold a reachability property with at least another node n_i that also holds this necessary condition.*
$Reachable(n_i,n_j) \Rightarrow Neighbor(n_i,n_j) \vee \exists n_1 \exists n_2 \exists n_3 \dots \exists n_x, x < |N| \mid (Neighbor(n_i,n_1) \wedge Neighbor(n_2,n_3) \wedge \dots \wedge Neighbor(n_{x-1},n_x) \wedge Neighbor(n_x,n_j))$.
Thus, it can be deduced that $Neighbor(n_i,n_j)$ corresponds to direct reachability. $Reachable(n_i,n_j)$ also eventually implies that there is a valid route (refer to Definition 2) from n_i to n_j.

A node $n_i \in N$ can perform the functions such as $request(n_i,n_j,Service)$, $response(n_j,n_i,reply)$, $forward(n_i,Msg)$, $discard(n_i,Msg)$, and $attack(n_i)$. $request(n_i,n_j,Service)$ denotes node n_i sends a request to node n_j for a particular *Service*. Service can be any network service or a request for sharing particular information. In our example, we have modeled the services such as *getExactLocation*() and *getAppoximateLocation*(). The service will reply the location of the requested node. After receiving a request $request(n_i,n_j,Service)$, the node n_j sends back the $response(n_i,n_j,reply)$ to n_i. The reply is the answer to the request. Reply might be encrypted or plaintext depending upon the service and policy of n_j. Any nodes $n_x \in R(n_i,n_j)$ should *forward* the network packet to another $n_y \in R(n_i,n_j) \wedge Neighbor(n_y,n_x)$. The network packet is originated at n_i and referred as *Msg*. Depending

on policy, bandwidth constraints, or selfishness of n_x, the *Msg* might be *dropped* instead of forwarding to n_y. *attack*(n_i) is the function that n_i executes to prove itself a malicious node or adversary node.

Definition 2 (Route). *Route $R(n_i, n_j)$ is defined as an ordered set of nodes that forward the network traffic from $n_i \in N$ to $n_j \in N$ if Reachable(n_i, n_j). From the definition of reachability, $\exists n_1 \exists n_2 \exists n_3 \ldots \exists n_x, x < |N| \mid (Neighbor(n_i, n_1) \wedge Neighbor(n_2, n_3) \wedge \ldots \wedge Neighbor(n_{x-1}, n_x) \wedge Neighbor(n_x, n_j))$, the ordered set $\{n_i, n_1, n_2, \ldots .n_x, n_j\}$ is the Route $R(n_i, n_j)$.*

Definition 3 (Trust of a Route). *Trust of a route is a quantitative measure of how trustworthy the route is. For a route $R(n_i, n_j)$, any member n_x of ordered set $\{n_i, n_1, n_2, \ldots n_x, \ldots n_j\}$ has a particular trust value with respect to n_i, $\tau(n_x, n_j)$. Set U represents the untrustworthiness of all the members of $R(n_i, n_j)$ with respect to its previous one, that is, $U = \forall n_x \in R(n_(i+1), n_j), (1 - \tau(n_x, n_(x-1)))$. Now, the trust of the route $R(n_i, n_j)$ is defined as $T_{R(n_i,n_j)} = 1 - (u_i \times u_i + 1 \times \ldots .u_j)$.*

Specification of network instance

The specification of the network instance is given to the framework through a specification language. The trust model gives the values of the trust matrix of the network instance. The keywords are *NI, QUAD, NODE, COMMENT, MOBILITY, TRUST, REACHABILITY, CONNECTIVITY*, and *TRUST*. *Parenthesis, brackets, semicolon*, and *comma* are used as delimiters. The network instance is specified as

```
NI allied_example1{
   QUAD q1 {
      NODE n1 {
        MOBILITY    random;
        TRUST       model1;
        REACHABILITY  auto;
        SERVICES serv1, serv2, serv5;
      }
      NODE n2 { ...  }
   ...
}
CONNECTIVITY
    ((1 1 0 1)(1 1 0 1)(0 1 1 1)
     (1 0 0 1));
TRUST
    ((1 .3 0 1)(.2 1 .6 .7)
     (0 .2 1 .6)(.4 .2 .3 1));
```

Reduction in Boolean model

The Boolean reduction process starts with mapping of the network instance into Boolean variables. The network instance has components such as quads, nodes, connectivity matrix, and trust matrix. If there are q number of quads, $\lceil \log_2 q \rceil$ bits are required to represent it. For our case, we have assumed 16 quads. So, four Boolean variables, namely q_3, q_2, q_1, q_0, are required to represent a quad. Similarly, if a quad has maximum 128 nodes, seven Boolean variables, $n_6 \ldots, n_1, n_0$, are used to represent a node. Function $\mathcal{FQ}(q3, q_2, q_1, q_0)$ represents a quad or a set of quads, and function $\mathcal{FN}(n_6, \ldots n_1, n_0)$ represents the nodes. So, $\mathcal{FNode} := \mathcal{FQ} \wedge \mathcal{FN}$ can uniquely represent a set of nodes in the network. A function represents a network resource; it means that it gives an output TRUE value if that resource exists and gives FALSE otherwise. Just as an example, if 0101 is a quad, then $\mathcal{FQ} \Leftrightarrow \neg q_3 \wedge q_2 \wedge \neg q_2 \wedge q_1$. Each of the services is represented by variables $r_0, r_1 \ldots r_7$. Boolean function $\mathcal{FServ}$ on the variables $r_0, r_2 \ldots$ represents services in the network. Connectivity matrix I is represented by a vector of Boolean variable of size n^2, where n is the total number of nodes. $\mathcal{FI}(p_0, p_1, p_2, p_3 \ldots .p_n)$ represents the connectivity matrix I, such that, value of i-th row, j-th column in the I is reflected in $p_{n.(j-1)+i-1}$. So, $Neighbor(n_i, n_j) \Rightarrow p_{n(j-1)+i}$. The function $\mathcal{FI} : \mathcal{FNodeXFNode} \rightarrow 0, 1$ is defined to check neighborhood of a node from $\mathcal{P}$. The underlying trust model gives the trust matrix T, as defined in Section 3. Just like the connectivity matrix, trust matrix is also represented by $\mathcal{T}(t_0, t_1, t_2, t_3 \ldots .t_n)$. The function $\mathcal{FT} : \mathcal{FNodeXFNode} \rightarrow 0, 1$ is defined to obtain the trust value of a node with respect to another from the $\mathcal{T}$. Output of $\mathcal{FT}$ is *TRUE* if trust value $\tau \leq C, 0 < C < 1$ and *FALSE* otherwise. This is how the network instance is reduced to Boolean clauses. Table I shows the procedure and an example for Boolean reduction of network.

For specifying the network, the syntax of the *network grammar* is described in Table II.

4. NETWORK ACCESS MODEL

Different policy rules are applied on network instance. Depending on the policy rules [13], a node takes the decision whether to reply to a request for a specific service or not. The access control on the network requires to be formally modeled for network access modeler to deduce the Boolean clauses. The network access model represents the policies applied on the network. Policy rules are written for the quads. It may correspond to a single node or a group of nodes. A policy rule P between a source node n_s and a destination node n_d defines whether n_d will respond positively for the request made by n_s for the service *serv*.

Definition 4 (Policy rule). *Policy rule P can be defined as four tuples, $\langle SRC, DST, Serv, Condtion \rangle$ along with an attribute Action $\in \{+, -\}$, where $SRC, DST \subseteq N$, and $request(src, dst, Serv) \wedge src \in SRC \wedge dst \in DST$. The condition is a Boolean function on the network instance. Condition must be true for a valid policy rule. All dst $\in$ DST executes for the aforementioned request according to attribute Action.*

Table I. Boolean mapping of the model.

Quad(Q) : $\mathcal{FQ}(q_0, q_1, ...q_4)$
Node(N) : $\mathcal{FN}(d_0, d_1, ..., d_6)$
Service(P) : $\mathcal{FP}(r_0, r_1..., r_7)$
ConnectivityMatrix(I) : $I(p_0, p_1, ...p_144)$
TrustMatrix(T) : $\mathcal{T}(t_0, t_1, ...t_144)$
Procedure Reduce_Network_Instance()
Input: Network Instance NI
Output: Boolean Reduction of Network Instance M
BEGIN
1. For all nodes
2. $Node \Leftrightarrow Node \vee (\mathcal{FQ}(q_0, q_1...q_4) \wedge \mathcal{FN}(n_0, n_1...n_7) \wedge \mathcal{FServ}(r_0, r_1..r_7))$
3. $I \Leftrightarrow \mathcal{FI}(p_0, p_1...p_144)$
4. $T \Leftrightarrow \mathcal{T}(t_0, t_1, ...t_{144})$
5. $G \Leftrightarrow Node \wedge I \wedge T$
6. Return M
7.END
Example: Boolean Reduction of a Network Instance
Input:
`NI g { QUAD q1 { NODE n1 { SERVICE serv1, serv2; ... } } QUAD q2 { NODE n2 { SERVICE serv2, serv3; ... }} } CONNECTIVITY: ((1 0)(1 1)) TRUST: ((1 .1)(.7 1))`
Output:
$q1 \Leftrightarrow \neg q_1 \wedge \neg q_0$; as, $q_1 = 00$
$q2 \Leftrightarrow \neg q_1 \wedge q_0$; as, $q_2 = 01$
$n1 \Leftrightarrow q_1 \wedge \neg n_1 \wedge \neg n_0$; as,$n_0 = 00$
$n2 \Leftrightarrow q_2 \neg q_1 \wedge q_0 \wedge \neg n_1 \wedge n_0$; as $n_2 = 01$.
$serv1 \Leftrightarrow \neg r_1 \wedge \neg r_0$
$serv2 \Leftrightarrow \neg r_1 \wedge r_0$
$serv3 \Leftrightarrow r_1 \wedge \neg r_0$
$N \Leftrightarrow n_1 \wedge \neg r_1 \vee n_2 \wedge (r_1 \wedge \neg r_0 \vee \neg r_1 \wedge r_0)$
$I \Leftrightarrow p_0 \wedge \neg p_1 \wedge p_2 \wedge p_3$
$T \Leftrightarrow t_0 \wedge \neg t_1 \wedge t_2 \neg t_3$;as $\tau(n_2, n_1) > C$, C is taken as 0.5
$M \Leftrightarrow N \wedge I \wedge T$
Note: Instead of too many variables, we took lesser variables in the example for simplicity.

$(P.Action = \text{`+'}) \Rightarrow response(dst, src, Serv) \wedge$
$(P.Action = \text{`–'}) \Rightarrow \neg response(dst, src, Serv)$.
Multiple conditions will be expressed as conjunction.

Table II. Network specification.

```
text            :=  [a-zA-Z][a-zA-Z0-9]*
number          :=  [0-9]+
%%
nw_instance     :=  NI nw_id {nw_declare}
nw_id           :=  text
nw_declare      :=  q_block r_block
q_block         :=  q_declare | q_declare
                    q_block
q_declare       :=  QUAD q_id {q_definition}
q_id            :=  text
q_definition    :=  adl_info n_block
adl_info        :=  COMMENT text ;
n_block         :=  n_declare | n_declare
                    n_block
n_declare       :=  NODE n_id {n_definition}
n_id            :=  text
n_definition    :=  adl_info n_info
n_info          :=  MOBILITY text;
                    TRUST text;
                    REACHABILITY text;
r_block         :=  con_statement tr_statement
con_statement   :=  CONNECTIVITY matrix ;
tr_statement    :=  TRUST matrix ;
matrix          :=  ( vectors )
vectors         :=  vector | vector vectors
vector          :=  ( elements )
elements        :=  number | number , elements
```

There are four types of condition in our model. These are as follows:

(1) *[Subsuming condition]:* The condition restricts the access on the basis of the source node that tries to request for a service. A policy rule action may depend on the containment of the requesting node in a specified set of nodes. For a policy rule P, a subsuming condition is modeled as $P.src \in N' \subseteq N$. Typically, the set N' is a quad or a group of quads. As an example of such subsuming condition, the policy rule might be

> Any node of Quad1 can access the exact position information of another node of Quad1. The Node n_3 of Quad2 can access exact information of any other nodes of Quad2. Other nodes of Quad2 are entitled to get the approximate location of the Quad2 devices.

This policy can be modeled as,
$P1 := \langle N_1, N_2, getExactLocation(), \forall n \in N_1, \forall n' \in N_2(n.quad == 1 \wedge n'.quad == 1)\rangle+$
$P2 := \langle N_3, N_4, getExactLocation(), \forall n \in N_3, \forall n' \in N_4(n.quad == 2 \wedge n'.quad == 2 \wedge n.id == 3)\rangle+$
$P3 := \langle N_5, N_6, getApproxLocation(), \forall n \in N_5, \forall n_6 \in N_2(n.quad == 2 \wedge n'.quad == 2 \wedge n.id! = 3)\rangle+$
All other rules are denied by default.

(2) *[Trust condition]:* The condition that controls the access on a service involving trust of the route or

trust of the requesting nodes falls under this category. There is an assumption that all the nodes have agreed upon a same trust model. Action of the policy rule $P_n\langle N_1, N_2, getExactLocation(), \forall n \in N_1, \forall n' \in N_2 T(n,n') \geq K\rangle$ allows the access from n to n' for service *getExactLocation*() if the trust of route between them has a value greater than K, where K is a positive real number. For our model, we have normalized K within the range of 0–10. The trust value of the requesting node can also impose condition for policy rules. $P_n\langle N_1, n', getExactLocation(), \forall n \in N_1, \tau(n',n) \geq K\rangle$ is such a policy rule. In our example, the policy "N3 and N4 is permitted to access exact location information of devices from Quad2 if there is a route with minimum trust value 5 and requesting node can have trust value below 4. Otherwise U1 and U3 can access only an approximate location of any UK device" can be modeled as follows,
$P4 := \langle N_1, N_2, getExactLocation(), \forall n \in N_1, \forall n' \in N_2 T(n,n') \geq 5 \wedge \tau(n',n) \geq 4\rangle+$. The other rules are by default deny.

(3) *[Event-based condition]:* Event-based condition controls the access to a service on occurrence of an event. Each event is represented by a Boolean function. If the event happens, the function becomes true. For example, the nodes of Quad3 (assuming it is the Red Cross Society) can access the service *getExactLocation*() only if a casualty happens. *IsCasualtyHappened*() is the function that returns Boolean *True* when casualty takes place in the area. The policy rule will modeled as,
$P5 := \langle n, n', getExactLocation(), n.quad == 3 \wedge \forall n' \in N \wedge IsCasualtyHappened()$

(4) *[Topology condition]:* Topology conditions restrict the access for certain spatial condition that affects the topology of the network. These kind of conditions are modeled on the basis of two functions, $neighbor(n,n')$ and $reachable(n,n')$. The policy can be framed like "n_5 will share its exact location with n_6 if n_3 is not in its radio range." This can be modeled as,
$P6 := \langle n_6, n_5, getExactLocation(), \neg neighbor(n_6, n_3)\rangle+$.
Another example of this category is "n_7 and n_8 can access approximate location of any device of Quad1 if the intermediate nodes are from Quad1 only." This policy is reflected in our model as
$P7 := \langle n_7, n_8, getApproxLocation(), \forall n \in R(n_7,n_8) n.quad == 1\rangle+$ Topology condition also can be modeled using Connectivity Matrix I of the network instance.

Condition of a policy rule P can be any combination of these types. Usually, conditions are written as conjunction unless specified as disjunctive clause. Instead of disjunctive conditions, we can split the policy rule into more than one. That is, policy $P\langle N_1, N_2, getExactLocation(), cond1 \vee cond2 \vee cond3 \ldots\rangle$ can be specified as,
$P' := \langle N_1, N_2, getExactLocation(), cond1\rangle$
$P'' := \langle N_1, N_2, getExactLocation(), cond2\rangle$
$P''' := \langle N_1, N_2, getExactLocation(), cond3\rangle \ldots$
The policy rules are implemented at the destination node *Dst*. So, the intermediate nodes in the route cannot cause violation in access control. We have assumed ID spoofing is not allowed in the network and the trust model on which the framework relied upon is sound enough to capture the adversary nodes. The policy rules, which are not specified, are by default deny rules, that is, for an unspecified policy rule P', $P'.Action =$ " − ".

Boolean modeling of policy rules

Node policy base $P_n(n_i)$ of a node $n_i \in N$ is a set of policy rules such that $\forall P, P.Dest = n_i \Leftrightarrow P \in P_n(n_i)$. Node permitted policy base $P_n^+(n_i) \subseteq P_n(n_i)$ of a node $n_i \in N$ is the set of policy rules for which, $\forall P \in P_n^+(n_i), P.Action =$ "+". Similarly, node denied policy base can be defined as $\forall P \in P_n^-(n_i) \subseteq P_n(n_i), P.Action =$ " − ". Quad policy set $P_q(Q_i)$ is $\forall n \in N \wedge n.quad = Q_1, \cup P_n(n)$. Similarly, we can define Quad Permitted Policybase $P_q^+(Q_i)$ and quad denied policy base $P_q^-(Q_i)$.

Definition 5 (**Consolidated policy model**). *Consolidated policy model M_c is the summation of all the policy rules. Consolidated permitted policy base P_c^+ may be defined as $\cup_{\forall Q} P_q^+(Q)$, and consolidated denied policy base P_c^- is $\cup_{\forall Q} P_q^-(Q)$. Consolidated permitted policy model M_c^+ is given by $\forall P_i \in P_c^+, \vee P_i$. Consolidated denied policy model M_c^- is $\forall P_i \in P_c^-, \wedge P_i$. If there are n numbers of permit rules in the model, $M_c^+ := (P_1^+ \vee (P_2^+ \vee \ldots P_n^+)$. Similarly, for n' number of deny rules, $M_c^- := (P_1^- \vee (P_2^- \vee \ldots P_n^-)$. Now, consolidated policy model M_c is defined as $(M_c^+ \wedge M_c^-)$.*

The policy rules may have conflicts and redundancy. The policy refinement is necessary for a correct and sound policy model. The conflicts can be analyzed in our framework. There might be intranode conflicts, internode conflicts, or allied policy conflicts. *Intranode conflict* is defined as $\exists P, P \in P \in P_n^+(n_i) \wedge P \in P \in P_n^-(n_i)$, whereas *inter rule conflict* is defined as $\exists P_1 \in P_n^+(n_1) \cap P_q^-(Q_i), \exists P_2 \in P_n^-(n_2) \cap P_q^-(Q_i), P_1.src == P_2.src \wedge P_1.dst, P_2.dst \in Q_i \wedge P_1.serv == P_2.serv \wedge P_1.condition = P_2.condition$. Inter rule conflict is a superset of intrarule conflicts as $P_1.dst \equiv P_2.dst \Rightarrow n_1 \equiv n_2$.

Definition 6 (**Allied policy**). *Allied policy A of a network instance NI is the set of high level specification of the behaviors or access control mechanism set by the corresponding authorities of the allied quads. The allied policy A can be modeled as a set of abstract policy rules P' that are not specific to a node. A can be specified in natural*

language or any higher level specification language. The corresponding parser will map the semantics to low level policy rules.

Now, A is the expected or desired behavior of the consolidated policy. M_c is the consolidated policy model. We need to check if $A \wedge M_c$ is satisfiable or not. We will find the unsatisfiable core. The unsatisfiable core contains the conflicting policies. If the Boolean model is satisfied, then the policies are nonconflicting. Thus, there is no need for conciliation.

Specification of policy rules

Policy is specified using a policy specification language. The syntax of specifying policy rule is < *Rule_No* >:< *Action* >< {*Src*} >,< *Dest* >,< *Service* > [,< *Condition* >];. {Src} and {Dst} are a set of nodes. The policy rule with multiple destinations is split into number of rules with single destination. As an example, R_1 : $DENY\{n_1, n-2\}, \{n_2, n_3\}, serv1$ is split as,
$R_1\_1 : DENY\{n_1, n-2\}, \{n_2\}, serv1$;
R_1_1 : $DENY\{n_1, n-2\}, \{n_3\}, serv1$. < *Service* > is a set of services provided by the destination node, as described in Definition 1. The *condition* > is an optional field.*SRC* and *DST* are two keywords used to represent source and destination as a set of nodes. Comparison operators (==, ! =, <, <=, >, >=) and logical operators (& (and), | (or), ! (not)) are used to express conditions. The relationship operator "." is used to express attribute of a node as, < *node* > . < *atttibute* >. As an example, $C1$: $SRC.quad == q2$ is a valid condition. Different functions such as trust(), trust_route, event_happened(), and isNeighbor() are used for representing and modeling policy. Different conditions are connected using *and*(&) or *or*(!) operators. *Not*(!) is a unary operator to represent negation of a condition. A policy rule would look like, R_1 : $\mathbf{DENY}\{\mathbf{n_1}, \mathbf{n_2}, \mathbf{n_4}\}, \{\mathbf{n_3}\}$, getApproxLocation, $\{n_1, n_2, n_4\}.quad == quad2 \& trust(SRC) > .5$ (Table III).

Boolean reduction of policy rule

The Boolean reduction of the network access model requires functional mapping of the rule components into Boolean clauses. Policy rule components include *A*ction, Src, Dst, Serv, Condition, sign, and cost. We model the *source* and *destination* with $m = \log q + logn$ number of bit each (in our example it is 7 bit), namely, $(s_0, s_1, \ldots, s_m)$ and $(d_0, d_1, \ldots, d_m)$, respectively. A set of nodes can be translated using disjunction ($\vee$) operator. We define the functions $\mathcal{FS} : \{s_0, s_1, \ldots, s_m\} \rightarrow \{0, 1\}$ and $\mathcal{FD} : (d_0, d_1, \ldots, d_m) \rightarrow (d_0, d_1, \ldots, d_m)$ to represent a set of source and destination nodes. Similarly, services are mapped into Boolean variables, namely, $(r_0, r_1, r_3 \ldots r7)$. $\mathcal{FR}(r_0, r_1, r_3 \ldots)$ represents the services. Modeling the conditions is challenging here. For subsuming conditions, we can add the constraint on the source or destination and can model the functions $\mathcal{FS}$ and $\mathcal{FD}$ in such a way that they incorporate the subsuming conditions. For event-based conditions, we have to take the variables $(e_0, e_1, e_3 \ldots, e_7)$. Different combination of these variables denotes different events. We define a function $\mathcal{FE}(e_0, e_1, e_3 \ldots)$ that represents the events happened. Using $\mathcal{FP}$, we can model the topology condition. For modeling trust-based conditions, we introduce two Boolean functions, namely, $\mathcal{FT}\_route$ and $\mathcal{FT}\_node$. $\mathcal{FT}\_route$ represents the trust of the route, and $\mathcal{FT}\_node$ denotes trust of the node. Variables $\left(t_0^r, t_1^r, t_2^r \ldots\right)$

Table III. Policy specification language.

Service(P) :FP($r_0, r_1, \ldots r_8$)
Src(SIP) :FS($s_0, s_1, \ldots s_1 1$)
Dst(DIP) :FD($d_0, d_1, \ldots, d_1 1$)
Condition(Cond) :FC($c_0, c_1 \ldots c_3 2$)
Action(g) : $A(g)$

Procedure Reduce_Rule()
Input: A Policy Rule r_i
Output: Boolean Reduction of the rule r_i
BEGIN
1. $P_i \Leftrightarrow FP_i(r_0, r_1, r_2..r_8) \wedge$
2. $SIP_i \Leftrightarrow FS_i(s_0, s_1, \ldots s_1 1) \wedge$
3. $DIP_i \Leftrightarrow FD_i(d_0, d_1, \ldots, d_1 1) \wedge$
4. $Cond_i \Leftrightarrow FC_i(c_0, c_1, .., c_{31}) \wedge$
5. $r_i \Leftrightarrow Serv_i \wedge SIP_i \wedge DIP_i \wedge Cond_i$
6. Return r_i
END

Example: Boolean Reduction of a rule

$\mathbf{Rule(r_1)}$:

ALLOW $\{n_1, n_3\}, n_2, serv1$ IF $trust(n_1) > .3$
$P_1 \equiv$ **serv1** $\Leftrightarrow (\neg r_1 \wedge r_0)$
$SIP_1 \equiv \{n_1, n_3\} \Leftrightarrow \neg s_0$; as $n_3 = 10, FS(s_1, s_0) = \neg s_0$
$DIP_1 \equiv n_2 \Leftrightarrow d_0 \wedge \neg d_1$
$Cond_i \equiv \{trust(n_1) > .3\} \Leftrightarrow \{\mathcal{FT}\_node(n_1) > .3\} \bigodot c_0$
$r_i \Leftrightarrow (P_1 \wedge SIP_1 \wedge DIP_1 \wedge Cond_1)$

Table IV. Boolean mapping of the policy rule.

policy	:=	policy_no auth sign
auth	:=	**<**src,dst,serv**>** | **<**src,dst,serv,cond**>**
sign	:=	**+** | **-**
policy_no	:=	**NUMBER**
src	:=	**(** sub **)**
dst	:=	**(** sub **)**
sub	:=	**NodeID** | **NodeID ,** sub
serv	:=	**getExactLocation** | **getApproxLocation** | ...
cond	:=	c_declare | c_declare **&** cond | c_declare **|** cond
c_declare	:=	c_statement | **!**c_statement
c_statement	:=	c_quantifier c_logic
c_quantifier	:=	**@ NodeID IN** sub | **#** **NodeID IN** sub | ϵ
c_logic	:=	attr op attr | attr op **NUMBER** | event
attr	:=	**NodeID.attribute**
op	:=	**>** | **>=** | **==** | **!=** | **<<=**
event	:=	**IsCasualtyHappened** | **IsSelfishNode** | **IsAttacker** | ... | **!** event

and $(t_0^n, t_1^n, t_2^n \ldots)$ are used to represent the Boolean range of the trust values. Now, the overall policy rule is modeled as $\mathcal{FS} \wedge \mathcal{FD} \wedge \mathcal{FR} \wedge \mathcal{FE} \wedge \mathcal{FT}\_node \wedge \mathcal{FT}\_route$. Table IV shows the procedure and elaborates it with an example.

Policy grammar would be as follows.

5. OPTIMIZATION POLICY IMPLEMENTATION

Policy implementation optimizer (PIO) unit of the framework (refer to Section 2) suggests the conciliation of the policy, which is required to achieve the coalition. All its possible implementation level has been assigned with weight. All those implementations along with the allied policy are modeled as a minimum-cost SAT problem [11]. This unit models the policy base as a minimum-cost SAT problem [11] and uses min-cost SAT solver for analysis. This is an iterative process for each of the conflicting policy rule.

We have defined a range of enforcement levels of priority for the policy rules. Each of the policy rule can be enforced in one of these levels. The corresponding administrators decide the priority level of the rule. Each of the services is classified according to approximation level. So, a service can be replied back with sharing the critical information fully or partly or none. As an example, if getLocation() is a service, getExactLocation(), getApproximateLocation(),noLocationInfo(), and so on are the approximation level of that service. The administrator of the quad determines these approximation levels of a service.

Definition 7 (Enforcement level of policy rules). *Enforcement level of a policy rule* $P \rangle N', n_i, SERV, COND\langle \pm$, *can be defined as a set of policy rules* $\left\{P_1^*, P_2^*, P_3^* \ldots P_n^*\right\}$ *such that* $P_1^* := P$ *and* $P_n^* :=\rangle N'$, $n_i, SERV\langle +$ *(all permit) and* $P_2^*, P_3^* \ldots P_{n-1}^*$ *are the policy with same source and destination but with varying approximation level of service (SERV) and strictness of condition (COND). Enforcement level of a policy can be written as* $\rangle N', n_i, SERV_i, COND_i\langle \pm$. *The administrator of the allied quads needs to categorize all the enforcement level of a policy rule. The administrator also needs to assign cost for each of the enforcement level.*

The cost of an enforcement level P_x^* is defined as the cost $c_x \in 1, 2, 3, 4, 5 \ldots 100$ that is associated to it. If P is a conflicting rule, $(P_1^* \vee P_2^* \vee P_3^* \ldots \vee P_n^*) \wedge A$ is the Boolean function ϕ. The goal of the model is to find a variable assignment $X \in \{0,1\}^n$ such that X satisfies the Boolean formula ϕ and minimizes $C = \sum_{i=1}^{n} c_i x_i$, where $x_i \in \{0,1\}$ and $1 \leq i \leq n$. In the SAT instance, the set of all the P_i^* for which corresponding variables in X are true is the desired conciliated policy. The process is repeated for each of the conflicting policy rules.

Table V. Grammar for policy enforcement level specification.

```
enforcement     :=  policy_no { policy_block
                    }
policy_block    :=  policy level; | policy
                    level ; policy_block
level           :=  NUMBER
```

Specification of policy enforcement level

For specifying policy enforcement, the administrator needs to write another set of policy rules. The grammar of the policy rules is the same as described in the previous subsection. Specification of the enforcement level of a rule < *RuleNo* >:< *ACTION* {*SRC*}, {*DEST*}, {*serv*}, {*Condition*} > is written as < *RuleNo* >: < *EnforcementLevel* >: < *ACTION* {SRC_i}, {*DEST*}, {$serv_i$} [, {$Condition_i$}]; >. Different enforcement level is achieved by different combination of services and conditions. The local administrator of the quad needs to provide these pieces of information (Table V).

6. EXPERIMENTATION

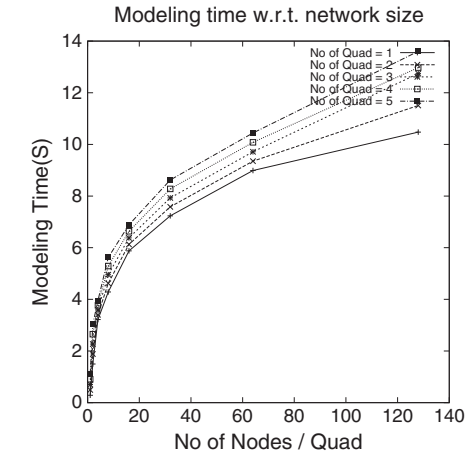
In this section, we show the performance analysis and feasibility studies of the proposed framework. We have tested our framework for more than 40 networks with varying loads and policy specifications. We focus our study on the time and space requirement for building and running the model. We study the effect of several parameters such as number of nodes, number of quads, number of policy rules, and enforcement levels of each policy rules. We have varied the parameters to thoroughly test the network. A random topology is generated as a mesh for the tests. We have built a random policy generator that generates a number of policy rules for the testing. zChaff [9] and its minimal-cost zChaff extension [11] are used for model checking. Modeling time to generate the CNF and SAT execution time is the main performance concerns. The number of variables and the number of clauses generated for the CNF affect on the space complexity, which is a major feasibility concern. Typically, for four quads, each with eight nodes in a network having total number of policy rules 100 and 5 enforcement levels, the parsing time is 0.009 s, SAT execution time is 0.005 s, the number of variables used is 232, and the number of clauses is 856. We have evaluated our framework in 50 different test networks with more than 1000 nodes and 1000 randomly generated security policies. In each cases, the conflict analysis and optimal solution generation times lie within 14 s. In addition, the maximum space requirement for modeling and analysis is 1.4 MB.

Impact of different parameters on the model is discussed in the succeeding text.

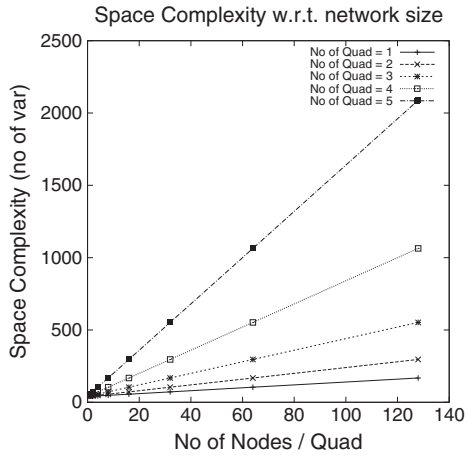
6.1. Impact of network size

Network size depends on the number of quads and the number of nodes. As we have modeled the nodes as

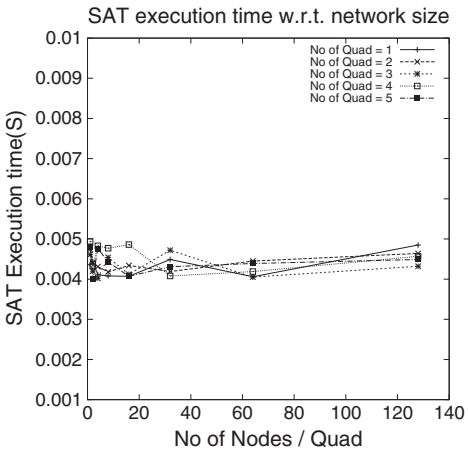
DOI: 10.1002/sec



(a) Modeling time with network size



(b) Space Complexity with network size



(c) SAT execution time with network size

Figure 2. Graphical plot of the analysis with network size.

Boolean variables, the n number of nodes can be represented by $\log_2 n$ Boolean variables; we have varied the number of nodes as a power of 2. The number of variables used for generating the CNF for SAT solver, the CNF generation time, and SAT execution time with varying number of nodes and quads is plotted in Figure 2a, 2b, and 2c, respectively. CNF generation time is polynomially increasing with network size as parsing time of the whole network increases for addition of each single bit. The curve has steps only at nodes or quads with a power of 2. Whereas SAT execution time is almost constant and space complexity becomes linear.

6.2. Impact of policy size

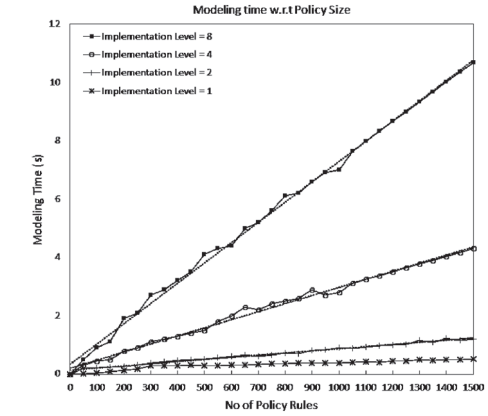
Policy size represents the summation of number of policy rules to each quads and the number of allied policies. Each of the policy rules is modeled into constant number of Boolean variables. So, the modeling time varies linearly with the number of policy rules. Figure 3a confirms the relation. This is to be noted that the simulation was run on a computer without controlling the runtime parameters. Thus, the points on the curves fluctuate slightly. The trend line shows the expected nature of the curve. For the same reason, in Figure 3b, we can see that the space complexity also linearly varies with policy rules. But SAT execution time does not follow any such pattern (Figure 3c), because it depends on the number of conflicting rules in policy base. As we have generated random rules, conflicts also occur in irregular pattern.

6.3. Impact of enforcement level

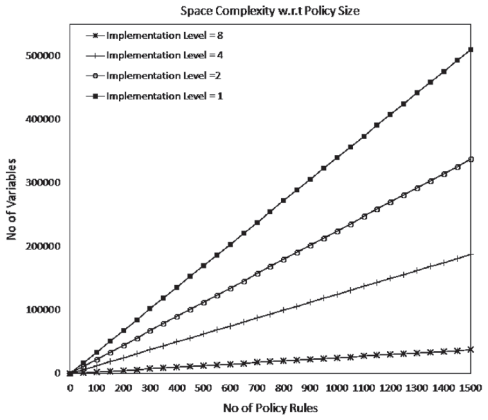
Each of the policy rules may have different enforcement level. The maximum number of level is taken as 10. The minimum number of level enforced is 1, that is, the rule cannot be compromised. Modeling the enforcement level maintains a linear proportion with policy rules (Figure 4a); whereas Figure 4b space complexity follows the similar pattern of parsing time, as parsing for each level is almost constant. Figure 4c shows the relation between SAT execution time and implementation level. It can be noted that for lesser number of enforcement level, it gives better performance, although minimizing the level increases the probability of unsatisfiability, causing no optimal solution.

7. RELATED WORK

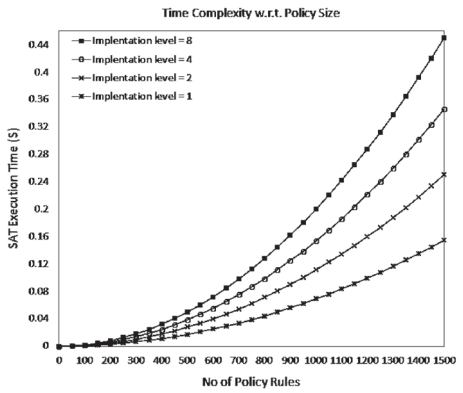
Policy-based security is a major need for such networks to protect the network resources from unauthorized accesses. A number of literature [14–16] focuses on access control in the individual resources in wireless and MANET. This access control enforcement becomes challenging in co-allied environment. Policy-based security for enterprise LAN was described by Bera *et al*. [17]. Access control in MANET is covered by Maity and Ghosh [5]. Ao and Minsky [18] introduced a model for coalition where local policies are prioritized according to global coalition policy.



(a) Modeling time with policy size

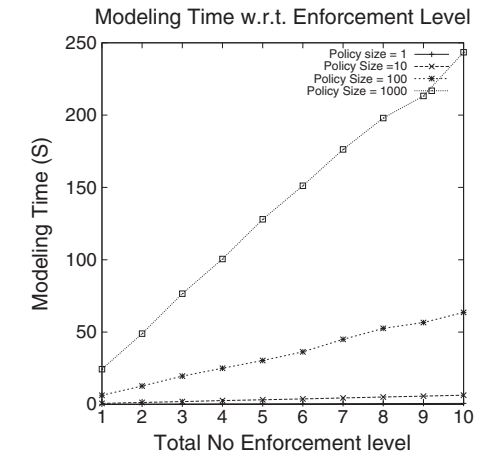


(b) Space Complexity with policy size

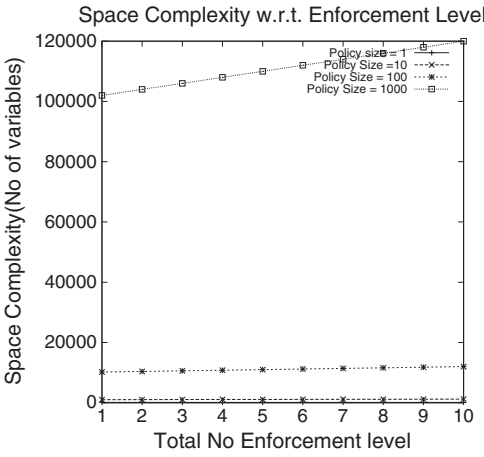


(c) SAT execution time with policy size

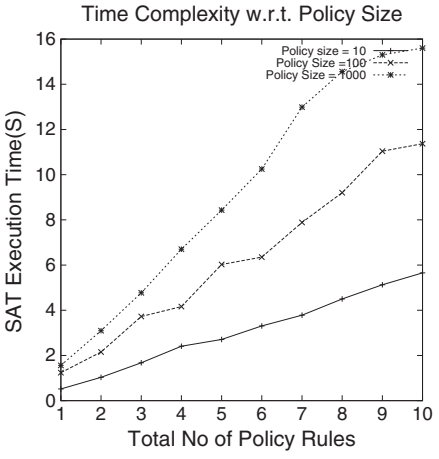
Figure 3. Graphical plot of the analysis with policy size.



(a) Modeling time with enforcing level



(b) Space Complexity with enforcement level



(c) SAT execution time with enforcement level

Figure 4. Graphical plot of the analysis with number of enforcement level.

Srivatsa *et al*. [19] introduced different aspects in security concerns in coalition of MANET. Zhao and Bellovin [20] has proposed a framework for the policy algebra. Their framework is capable to handle policy in coalition MANETs. To meet the global security policy requirements, the policy algebras provide a formalism to compute different operations such as summation, conjunction, and subtraction on the rule sets. Their framework also modeled the cost and risk associated with policy enforcement. Policy outsourcing tries to ensure global cost minimization. Access control policies have a very clear and

DOI: 10.1002/sec

restricted semantics over a collection of subjects, objects, and action terms. Wijesekera and Jajodia [21] presented a different approach of policy algebra using propositional logic for access control. They model policies as nondeterministic relations over a collection of subjects, objects, and action terms. Unlike the policy algebra framework, we model the enforcement of a rule in different levels and find the optimal solution. Janicke *et al*. [16] proposed a dynamic access control policy algebra in a recent work. However, Bandra *et al*. [22] proposed a scheme for policy refinement for Internet protocols. Recent work by Zhao *et al*. [13] focused on policy refinement in coalition MANETs. But they have not proposed any solution for resolving the conflicts between local and global policy rules. In recent days, Lv *et al*. [23] had proposed a scheme for policy optimization using secure group key communication. Our approach provides a formal validation of the framework, which is not done by their negotiation protocol. Taleb and Hadjadj-Aoul [24] had proposed a scheme to ensure quality of service after policy integration in MANET. As our system is an offline module, it does not need such protocols.

Beigi *et al*. [25] and Chen *et al*. [26] proposed a negotiation framework for coalition policies in MANET. Their work is mainly based on an automated negotiation schemes using rule taxonomy. Their work is based on the software negotiation protocol introduced by Bartolini *et al*. [27]. The proposed framework by Beigi *et al*. enables different network entities to collaborate and negotiate their offers in reaching a mutual goal. Our model is different than negotiation as we are trying to find the optimized enforcement of the conflicting policies instead of negotiating it. And optimization adds the efficacy to our work.

In management field, Wall and Callister [28] had proposed the method of conciliation in a business deal. Billing [29] presented the conciliation and conflicts in the computer mediated communications in the field of human-computer interface. In recent works, Enserink [30] has given a policy analysis model for conciliation. We propose conciliation for the MANET policy in coalition environment that eventually gives the optimal solution, by which the quads need to compromise the minimal level of stringency in the enforcement level of their local policies to achieve the successful allied network.

8. CONCLUSION

In heterogeneous networks like co-allied MANET, because of the agreements between the allied teams (quads) and enforcement of their existing security policies, conflict may occur. This paper presented a formal framework, named PoliCon, for finding optimal conciliated policy enforcement in co-allied MANET. In such mission critical networks, one of the major concerns is negotiating the security policies among allied parties. Instead of negotiating with policies, PoliCon framework establishes a novel approach of conciliation for compromising with the enforcement of the policy to the optimal level toward a successful coalition. However, in co-allied MANET environment, enforcement of these security requirements may conflict with allied policies framed by the mutual agreements between the co-allied parties (quads). In addition, these requirements may be temporal or static with fine-grained constraints (trust, mobility, etc.). This paper presented a formal framework for finding optimal conciliated policy enforcement in co-allied MANET. This framework formally models the network instances considering various nodes with their properties into Boolean logic. By reducing the local policy requirements of different nodes and the allied security policy under the same formalism, the framework verifies for conflicts between the local and allied policy constraints using zChaff SAT solver. Finally, the framework uses minimal-cost SAT analysis to find the optimal solution form of the model. The time complexity varies linearly with policy and network size. Incorporating mobility models with the framework and managing temporal policies and field study of the implemented framework are the potential future directions of this work.

REFERENCES

1. Luo H, Kong J, Zerfos P, Lu S, Zhang L. Ursa: ubiquitous and robust access control for mobile ad hoc networks. *IEEE/ACM Transactions on Networking (ToN)* 2004; **12**(6): 1049–1063.
2. Alicherry M, Keromytis AD. DIPLOMA: distributed policy enforcement architecture for MANETs, *Fourth International Conference on Network and System Security*, Melbourne, Australia, 2011; 89–98.
3. Von Mulert J, Welch I, Seah WKG. Security threats and solutions in MANETs: a case study using AODV and SAODV. *Journal of Network and Computer Applications* 2012; **35**(4): 1249–1259.
4. Maity S, Bera P, Ghosh SK. An access control framework for semi-infrastructured ad hoc networks, *2nd International Conference on Computer Technology and Development (ICCTD)*, Cairo, Egypt, 2010; 708–712.
5. Maity S, Ghosh SK. Enforcement of access control policy for mobile ad hoc networks, *Proceedings of the Fifth International Conference on Security of Information and Networks*, Jaipur, India, 2012; 47–52.
6. Maity S, Ghosh SK. A cognitive trust model for access control framework in manet. In *International Conference on Information Systems Security*, vol. 7671, Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2012; 75–88.
7. Merkel S, Mostaghim S, Schmeck H. A study of mobility in ad hoc networks and its effects on a hop count based distance estimation, *2012 5th*

International Conference on New Technologies, Mobility and Security (NTMS), Istanbul, Turkey, 2012; 1–5.

8. Bettstetter C, Resta G, Santi P. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing* 2003; **2**(3): 257–269.
9. Fu Z, Marhajan Y, Malik S. *Zchaff*. Research Web Page. Princeton University: USA, (March 2007). Available from: http://www.princeton.edu/˜chaff/zchaff.html. Accessed on November 2013.
10. Dillig I, Dillig T, McMillan KL, Aiken A. Minimum satisfying assignments for SMT. In *Computer Aided Verification Computer Aided Verification*. Springer: Berlin Heidelberg, 2012; 394–409.
11. Li XY. *Optimization Algorithms for the Minimum-Cost Satisfiability Problem*. North Carolina State University: Raleigh, USA, 2004.
12. Pirzada AA, McDonald C. Establishing trust in pure ad-hoc networks. In *Proceedings of the 27th Australasian Conference on Computer Science*, Vol. 26. Australian Computer Society, Inc.: Darlinghurst, Australia, 2004; 47–54.
13. Zhao H, Lobo J, Roy A, Bellovin SM. Policy refinement of network services for MANETs, *The 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011)*, Dublin, Ireland, 2011; 113–120.
14. Chadha R, Cheng H, Cheng YH, Chiang J, Ghetie A, Levin G, Tanna H. Policy-based mobile ad hoc network management, 2004.
15. Maity S, Bera P, Ghosh SK. A mobile IP based WLAN security management framework with reconfigurable hardware acceleration, *Proceedings of the 3rd International Conference on Security of Information and Networks*, Rostov-on-Don, Russia, 2010; 218–223.
16. Janicke H, Cau A, Siewe F, Zedan H. Dynamic access control policies: specification and verification. *The Computer Journal* 2013; **56**(4): 440–463.
17. Bera P, Ghosh SK, Dasgupta P. Policy based security analysis in enterprise networks: a formal approach. *IEEE Transactions on Network and Service Management* 2010; **7**(4): 231–243.
18. Ao X, Minsky NH. Flexible regulation of distributed coalitions, *Proceedings of 8th European Symposium on Research in Computer Security*, Gjøvik, Norway, October 13-15, 2003; 39–60.
19. Srivatsa M, Agrawal D, Balfe S. Bootstrapping coalition manets. *IBM Research Report RC24588*, 2008.
20. Zhao H, Bellovin SM. Policy algebras for hybrid firewalls, *Annual Conference of ITA (ACITA)*, New York, USA, 2007; 74–77.
21. Wijesekera D, Jajodia S. A propositional policy algebra for access control. *ACM Transactions on Information and System Security (TISSEC)* 2003; **6** (2): 286–325.
22. Bandara AK, Lupu EC, Russo A, Dulay N, Sloman M, Flegkas P, Charalambides M, Pavlou G. Policy refinement for IP differentiated services quality of service management. *IEEE Transactions on Network and Service Management* 2006; **3**(2): 2–13.
23. Lv X, Li H. Secure group communication with both confidentiality and non-repudiation for mobile ad-hoc networks. *IET Information Security* 2013; **7**(2): 61–66.
24. Taleb T, Hadjadj-Aoul Y. QoS2: a framework for integrating quality of security with quality of service. *Security and Communication Networks* 2012; **5** (12): 1462–1470.
25. Beigi M, Lobo J, Grueneberg K, Calo S, Karat J. A negotiation framework for negotiation of coalition policies, *2010 IEEE International Symposium on Policies for Distributed Systems and Networks*, Fairfax, USA, 2010; 133–136.
26. Chen K, Qiu X, Yang Y, Rui L. Negotiation-based service self-management mechanism in the MANETs, *2011 13th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Taipei, Taiwan, 2011; 1–7.
27. Bartolini C, Preist C, Jennings NR. A software framework for automated negotiation. *Software Engineering for Multi-Agent Systems* 2005; **3**: 213–235.
28. Wall JA, Callister RR. Conflict and its management. *Journal of Management* 1995; **21**(3): 515–558.
29. Billings MJ. Conflict, conciliation and computer-mediated communication: using online dispute resolution to explain the impact of media properties on relational communication, 2008.
30. Enserink B, Koppenjan JFM, Mayer IS. A policy sciences view on policy analysis. In *Public Policy Analysis*. Springer: Berlin Heidelberg, 2013; 11–40.