Improving Secure Communication Policy Agreements by Building Coalitions

Srilaxmi Malladi¹, Sushil K. Prasad¹, and Shamkant B. Navathe²

¹Georgia State University Dept. of Computer Science Atlanta , GA 30303 USA {cscsrmx,sprasad}@cs.gsu.edu

Abstract

In collaborative applications, participants agree on certain level of secure communication based on communication policy specifications. Given secure communication policy specifications of various group members at design time, the minimum set of resources for a pair, called Resolved Policy Level Agreement (RPLA) is translated into appropriate security service implementations, for the pair-wise communication to take place. We propose a novel idea that the members may extend pair-wise communication quality through other trusted nodes whose communication resources offer more security. We propose a heuristic algorithm which finds the best quality of protection (QoP), a measure of the resistance to an attack, path through coalition of trusted nodes. The results from our experiments indicate a significant improvement in QoP in the range of 13% to 48% over pair-wise communications.

Keywords: coalition policies, communication strength, secure communication, widest paths

1. Introduction

With the growing networking technology, there are numerous e-applications that are evolving. These range from very simple game applications to corporate decision making applications like programmed trading, real time stocks, military, etc. In a collaborative environment, secure communication among group members plays a crucial role for a successful collaborative application [7]. It is usual to agree upon a set of communication specifications to determine the required security services among source-destination pairs before initiating information flow. Each member manifests ²Georgia Institute of Technology College of Computing Atlanta, GA 30332-0280 USA sham@cc.gatech.edu

its communication requirements in terms of communication policy¹ [10]. The communication requirements among all collaborative members may not be symmetric and the members may have different communication resource capabilities.

We draw an analogy between the secure communication policy specification and the Quality of Service (QoS) specification. The policies in QoS refer to the service level attributes like bandwidth, delay, jitter, etc. The secure communication policy specifications refer to the hardware and software implementations for a secure communication and thereby the communication strength. A secure communication specification includes the choice of communication speed, route constraints, QoS parameters, firewalls, passwords, communication medium (wireless/wired), cryptographic mechanisms, access control, remote/local domain, encryption strength, authentication, etc. For any communication route, the medium of communication also plays an important role besides QoS attributes. Wireless networks are more susceptible to attacks like interference from portable phones and weakening of signal strength than wired networks [20]. We define Quality of Protection (QoP) as a numerical value representing resistance to an attack on a communication path. Thus, communication route over those links providing better QoP is less vulnerable to intruder attacks.

There is some effort to measure QoP [2, 13]. In [2], the collaboration-protocol profile and agreement specification describes a technique for mutually-acceptable quality of protection for exchanges between a service provider and a service consumer. Similarly, in military coalitions this is given as RPLAs. The authors in [13] give assurance measure in the form of Quality of Protection to constrain the information flow among the nodes to minimize risk.

Resolved Policy Level Agreement (RPLA) defined

^{1-4244-0910-1/07/\$20.00 ©2007} IEEE.

¹In this paper, reference to security is assumed whenever we use the phrases "communication" or "policy".

for pair-wise communication is obtained as the minimum of intersections from specifications and bindings/implementations of participating members. [14]. In large coalitions, it is likely that pair-wise intersection of all members' communication capabilities does not necessarily exist resulting in null sets. One way to resolve the issue of null intersection for RPLAs is to revise communication policies to guarantee a non-null RPLA. However, this process is both time consuming and infeasible in ad-hoc applications like military, conferencing, etc. We therefore extend the idea of RPLA to improve both the reach of communication and the level of security among members by bridging the gap in the existing pair-wise communication through trusted intermediary nodes.

Motivation: In ad-hoc coalitions, due to the nature of the network, there is a possibility that some of the existing communication mechanisms may have been damaged/lost during the application and, also that different members have varying communication resources at different states of application. Members may communicate via a direct pair-wise communication or an indirect communication through an intermediary member. Just as the presence of intermediate nodes guarantee that the communication does not halt even when communication network is partially connected or disconnected for various network mobility and connectivity reasons, and the alternative possibly longer routes can achieve better network QoS, the thrust of our work is the notion that longer paths can not only enable secure communication when not possible directly, but can also possibly enhance the level of security due to better encryption using longer keys, access control, authentication/filtration, distributed trust for digital certificate authentication, etc. This is now beginning to be recognized in the research community (see Section 5). For example, [5] formulates specific metrics for calculating the level of security along a path of intermediaries for three separate security aspects: network router neighborhood authentication, encryption, and access control. While encryption strength of a path is determined by its weakest link (a concave parameter), the trust in a message can build up with the number of nodes it has successfully passed through because of the access control filtration mechanisms it has been subjected to (a multiplicative parameter). On the surface, the idea of intermediate nodes seems to elevate insider and outsider attacks. However, with well defined trust models and careful selection of intermediate nodes, the additional nodes are not a threat and may even improve the communication quality and computation reliability [11, 15, 21, 24].

Given an initial pair-wise communication, finding maximum QoP subject to including trusted intermediate nodes is the goal of this paper. We represent communication policy as a 4-tuple: (*source, destination, Rsrcs, Actn*). The source and destination are the two end parties communicating, *Actn* (action) is the service attribute for which the intermediate node is being used such as authentication, authorization, QoS, encryption, etc., and *Rsrcs* (resources) are different sets of algorithms or bindings needed to carry out the specified attribute.

We describe QoP based on the way the intermediate node and link resources are used for enhancing the security of collaborative communication. The computation and communication efficiency varies for various combinations of intermediate nodes' QoS values and the amount of resources that can be expended for security services. For example, two devices communicating through a wireless medium can elevate the level of QoP by choosing to communicate through a trusted wired server. Similarly, when the information is delivered across a low speed communication link versus high speed communication link, different resource levels are needed.

The communication network among the coalition members is represented as a graph structure G = (V, E), where the vertex set for *n* members is $V = \{v_1, v_2, ..., v_n\}$ and the edges are communication links among these vertices. In our coalition system, resources for an *Actm* (action) may be asymmetric for a given pair of source and destination. We define the capability of each vertex v_i as $c(v_i) = \{r_1, r_2, ..., r_k\}$, where r_j , $1 \le j \le k$, is a resource capability at node v_i , such as strength of encryption/access control mechanisms, battery power, bandwidth, trust, computational resources, and other node and link attributes. Node attributes are usually implementations, algorithms, or techniques for a particular security service, where as link attributes are the QoS properties such as speed, bandwidth, link type, etc.

Without loss of generality, throughout the paper, we take encryption as *Actn* and strength of encryption as *Rsrcs*. For example, as proposed in [5], one could define the encryption metric based on encryption key length. The authors in [5] measure the degree of vulnerability (DoV), a value between zero and one, to denote a secure link or not depending on if the encrypted data sent over a link using a key is breakable or not for the next 30 years, defined as:

$$W_P = Min\{W_1, W_2, \cdots, W_n\}$$

$$W_i = \begin{cases} 1 & \text{if key size used is < the suggested size} \\ 0.99 - 0.033 * Y & \text{Y} = \text{no. of years the key is unbreakable} \end{cases}$$

 W_P is DoV along entire path and W_i corresponds to the *i*th node. The strength of encryption algorithms can also be compared based on the time, effort and human/machine resources needed to compromise an encryption algorithm for the same amount of data and key length [9, 12, 27]. The encryption metric is concave, i.e., it is upper bounded by the minimum over all the links in a path. The other routing

parameters may also be categorized as multiplicative and union. We define a function *strength* associated with each resource r_k as $s(r_k)$ to return an integer representing strength of encryption resources. The common capability list between nodes v_i and v_j is given by $c(v_i) \cap c(v_j)$. The *quality of communication* over an edge $(v_i, v_j) \in E$ is : $q(v_i, v_j) =$ max $\{s(r_k) | r_k \in c(v_i) \cap c(v_j)\}$, $(q(v_i, v_j)$ is otherwise negative infinity if $\{v_i, v_j\} \notin E$). For a path P in communication graph G containing a sequence of edges, QoP(P) = min $\{q(e) | e \in P\}$. QoP(P) is obtained as minimum value since it is a concave metric. For simplicity, no loops are considered in the network graph.

Limitations of earlier work: The current state-of-theart confines the communication policy bindings to pairwise. In [14], the authors obtain a communication mechanism based on the agreeable set of member policies from their initial pair-wise comparison. As the communication resources and the participants change in an ad-hoc manner, we need to effectively extend coalition policies. With current coalition communication schemes, the communication may halt if there is no common end-to-end communication scheme or the existing mechanisms disrupt. Related literature and comparison with our work is further presented in Section 5.

Contributions:

- We propose a novel graph representation for communication security policy.
- We enhance communication security among coalition members by introducing peer trusted intermediate nodes and extend the communication reach.
- We introduce Quality of Protection (QoP) as a routing parameter in secure communication.
- We give a heuristic algorithms to find an efficient communication route in network graph that maximizes the QoP using *trusted* set of intermediate nodes given as a hierarchy graph. It is common to have a hierarchical organization of certificate authorities, for example, therefore not all peers can be equal members as intermediaries.
- Our results on the proposed algorithms indicate an improvement in the average QoP value in the range of 13% to 48% without incurring significant overhead time.

The rest of the paper is organized as follows: Section 2 expands the problem statement and gives a naive solution, Section 3 presents a graph based algorithm to enhance communication security among coalition members, Section 4 discusses the implementation and performance results, Section 5 gives related work, and Section 6 gives our conclusions.

2. Problem Statement

We aim to maximize QoP in a communication route where only trusted nodes serve as intermediary functional nodes for enhanced security. An active trusted intermediate node may additionally perform the requested security services like encryption, authorization, etc., if needed. Any non-trusted/inactive trust node serves only as a forwarding node without performing the security services. The active nodes and inactive nodes are useful in applications like certificate chain discovery, energy efficient communications, etc. There could be many variations to define trust on the intermediate nodes; all members in the network graph are equally trusted (complete peer-to-peer trust), numerical trust value computed using trust models [22], proximity nodes, etc. We can extend the proximity idea to common ancestors of source and destination at different levels in trusted hierarchy graph and similarly to k-hop in a network graph.

The maximum QoP objective function gives the widest path of QoP, analogous to maximum bandwidth problem in QoS [17]. The widest path is usually computed for concave metrics that become bottleneck value over the entire path, such as bandwidth and QoP. The widest paths are optimal and can be found by choosing the best available optimization of a single criterion or a scalar function of many criteria [23]. The shortest widest paths are NP-hard as satisfying multiple criteria is proven to be NP-hard [4]. We therefore provide a heuristic solution for the QoP path computation problem. If we have multiple paths with maximum QoP value available for the same source-destination pair, we choose the one with shortest path among these. Shortest path is achieved when there is no other path that can yield a better optimization value by including fewer edges (nodes) than the claimed shortest path. Note that the objective function for a communication policy are specific to context in Actn and the associated rsrcs. For example, if an Actn in policy tuple is to minimize certificate discovery route, then we aim to find a route with resources that minimize certification path.

Assumptions: We make the following assumptions on the coalition system:

- We assume that the strength of various security services can be quantified by an integer. The quantification of security implementation algorithms is not currently prevalent but there are efforts underway. For example, an integer may be computed assuming that encryption algorithms are relatively comparable based on the time, effort and human/machine resources needed to compromise an encryption algorithm for the same amount of data and key length [9, 5, 12, 27].
- We are willing to trade off cost and time overheads for secure routing. Such trade offs are practical for

most non real-time applications involving sensitive data streaming and users willing to pay for security.

 Coalition members have a peer trust to achieve the common goal. Coalition members are willing to serve as intermediary members on a communication route from source to destination.

A set-based solution: A naive solution for the QoP path computation problem is to prune all sourcedestination pairs along the communication sequentially. Applying set algebra with the 4-tuple set notation (*source, destination, Rsrcs, Actn*), we can obtain maximum QoP computation for a single source and single destination. The idea of set algebra springs from the policy algebra proposed in [26], which is a collection of operators to combine different access control policies, and a collection of propositions to model conditionals.

In this approach, we assume that all members have a peer-to-peer trust approach and Rsrcs(i) denote the resource set available at vertex *i*. The non-empty intersection of Rsrcs for an Actn between different set pairs gives the common resource set available for a given source-destination pair. As QoP is a concave parameter, we take the minimum among the union of all common resources (obtained by intersections) along the path from every source to destination to obtain the maximum QoP value in the coalition. The operations of unions and intersections are done recursively for all source and destination pairs to identify the best set of intermediate nodes, leading to an exhaustive computation. This approach gives rise to overlapping sub-problems for each source-destination pair. The sub-problem solutions can be retained and used for maximum QoP computation for a coalition. This makes us believe dynamic programming technique can be very effective for the maximum QoP path computation and reduce the exponential complexity.

3. Graph Approach to the Problem

The overlapping sub-problems in the aforementioned set approach suggest dynamic programming on a graph as an efficient solution. The sub-problems identified in set approach show the existence of recursive relationship to define the optimal decision for iteration i, given that iteration i-1 has already been solved.

3.1 Communication Network Graph

The problem for the graph approach can be formally stated as:

Input: Given network graph G and hierarchy graph H with communication resource strengths associated on the edges between vertices of graph G. H defines the trust among members; the members high in the hierarchy are

trusted more and assumed to have more efficient resources.

Output: The list of vertices in the best (suboptimal) QoP path for all source - destination pairs.

The intermediate vertices in QoP path should be trusted to maintain the security and integrity of coalition. In this paper, trust is defined from the administrative hierarchical relationship with respect to both source and destination, given as hierarchy graph H. This follows that a member in lower hierarchy (M_1) obeys a member in higher hierarchy (M_h) by the rule of hierarchy (say administrative hierarchy) [3]. With a similar reasoning, source and destination nodes trust all the Common Ancestors (CA) in the hierarchy. Hierarchy graph therefore can be thought of as an overlay graph defining 'qualified intermediate nodes' and the heuristics are applied only on these nodes if all nodes are not trusted peers. Communication network graph G can be directed or undirected depending on whether the communication along links is asymmetric or symmetric, respectively. In an undirected graph, the source v_i and destination v_i can communicate with the same resource mechanisms both ways, i.e., $q(v_i, v_j) = q(v_j, v_i)$. Maximum QoP between v_i and v_j is denoted as $Q(v_i, v_j)$. Q_{ij}^k is defined as the best QoP path between v_i and v_j with trusted intermediary nodes in $\{1, 2, ..., k\}$. We define T_{ij}^k as the set of nodes trusted by both nodes v_i and v_j and therefore is the set of 'qualified intermediate nodes' for communication along (v_i, v_i) . The formalization can be extended to directed graph where the communication along an edge is not symmetric; the resource set of v_i to v_i and v_i to v_i are different i.e., $q(v_i, v_j) \neq q(v_j, v_i)$.

If hierarchy graph H and communication graph G have the same vertex set, and $E(H) \subseteq E(G)$, the QoP path from source to destination in G can be obtained as the path in hierarchy H. If H has a subset of vertex and edge sets of G, some pre-processing is done to ensure the inclusion of only trusted intermediate nodes. We expand details of pre-processing in Section 3.2.

3.2 Formalizing the Solution

The communication can be targeted from any source to any destination in the coalition. Any such path can be categorized as an all-to-all path. Floyd-Warshall algorithm employs dynamic programming technique and computes allto-all path. We employ Floyd-Warshall's algorithm as the basis of our heuristic solution to compute maximum QoP path, with an additional pre-processing step that finds the set of trusted intermediate nodes. A straight forward dynamic programming approach is not valid since the order in which vertices connect differ in H and G and may not guarantee valid sub instances as is, without any pre-processing. We

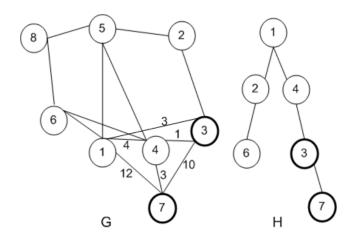


Figure 1. Communication graph G and Hierarchy Graph H

need an additional step to make sure all the sub-instances computed are valid in the next iteration. We retain only those edges that are reachable through trusted members as defined by H and apply heuristics only on such edges (other edges are not trusted even for routing). We refer to this version of algorithm as 'secure routing.' For applications such as digital certificates for authentication services, all edges may be trusted for routing and only some nodes are trusted for certificate issues and checks. In such applications, we can extend the connectivity using non-trusted nodes. We refer to this version of algorithm as 'secure computing.'

As shown in Figure 1, for source (3) and destination (7) and only common ancestors to trust, 1 and 4 are the only possible intermediaries. However to compute possible intermediate path 3-4-7, we again follow common ancestor approach on each path segment. For example a path between node 3 and node 4 has many possibilities: 3-7-1-4, 3-2-5-4, etc. Since 3 and 4 can trust only 1 as an intermediary (from H),we cannot choose any path other than 3-1-4 for a trusted secure communication. The graph H in Figure 1 is taken as binary tree only to keep the discussion simple. In reality, H can be of any hierarchical structure and does not affect our proposed Algorithm BestQoPPaths. The experimental results are presented in Section 4.

The algorithm to compute QoP in undirected network graph G is presented in Algorithm BestQoPPaths. The algorithm is done in two phases. In Phase I, trusted nodes can be computed for all source-destination pairs as part of preprocessing using a modified post-order traversal of H and stored in a boolean matrix T. The $T_{i,j}^k$ value is 1 if k can trusted as an intermediate node between i and j. The time complexity for post-order traversal is O(n). For computing trusted nodes for an *i*-j pair of nodes $(T_{i,j}^k)$, an O(n) time is needed for pruning all nodes and there are $O(n^2)$ source-destination pairs in total, making the overall time complexity $O(n^3)$. The QoP update statement in the Algorithm BestQoPPaths is valid as the intermediate nodes for the path segments Q_{ik} and Q_{kj} are obtained using phase I and are therefore available. The computation of QoP can also be considered based on the number of edges to be included, equivalent to m-hop constrained shortest path.

Algorithm : BestQoPPaths

INPUT: A network graph G and hierarchy graph H OUTPUT: All pairs shortest-widest QoP path with common ancestors as intermediaries

Phase I: Compute Common Ancestors (CA) Use modified post-order traversal of Hierarchy Graph H to compute the set of descendant (desc) nodes in H: (i) For each node k in graph H

- $desc(k) = k \cup desc(children(k))$
- (ii) For all i, j in desc(k), $T_{i,j}^k = 1$, else $T_{i,j}^k$ remains 0 $(T_{i,j}^k = 1 \Rightarrow k \text{ is CA of i and j})$

Phase II: Maximum QoP in Undirected graph and using Common Ancestors as intermediaries

$$n \Rightarrow |V|, Q_{ij}^{0} \Rightarrow q(v_{i}, v_{j})$$

for k = 1 to n do
for i = 1 to n do
for j = 1 to n do
if $(T_{i,j}^{k} = 1)$ then
if (ik \in E and kj \in E) then
 $Q_{ij}^{k} = \max\left\{Q_{ij}^{k-1}, \min\left\{Q_{ik}^{k-1}, Q_{kj}^{k-1}\right\}\right\}$
end for
end for
return Q^{n}

Algorithm BestQoPPaths can be easily modified to compute maximum QoP path for an undirected graph when all the neighboring nodes in network graph G can be trusted, referred as p2p path in our subsequent discussion. This trust criterion avoids the pre-processing step and is simply equivalent to either initializing $T_{ij}^{K} = 1$ for all k or ignoring the if condition (if $T_{i,j}^{k}$ is 1) in Algorithm BestQoPPaths when computing QoP in Phase II.

Time complexity analysis: The time complexity to embed group (of size n) policy tuple specification into a graph structure for n^2 pairs is $O(n^2)$. The actual QoP computation is done based on Floyd-Warshall's shortest path algorithm whose time complexity is $O(n^3)$. The algorithm is applicable for small and medium groups and may not be very well scalable for large group. The complexity may be further reduced by limiting the number of hops that are computed by

either comparing to a user defined satisfaction level or until no further change is seen in further iterations. One can extend the idea to distributed algorithms (online) by making each node independently to be able to compute its own 'quality path.'

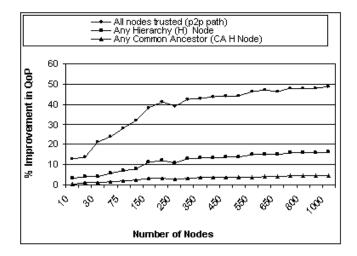
The proposed algorithm can handle computations for any coalition change. An addition to existing coalition is implicit by the nature of algorithm, while for deletion a deleted list can be maintained for efficiency. We consider current path through intermediaries, only if the intermediate node is not in deleted list. If there is a permanent deletion, we recompute only the affected nodes. The root node does not become the bottle neck because the communication quality offered is different at various tree levels.

The correctness of Algorithm BestQoPPaths can be proved based on the correctness of Floyd-Warshall's algorithm. Due to space constraints we omit the discussion on the algorithm proofs. Interested readers may refer to [16] for detailed discussion on the proofs.

4 Performance Evaluation

In this section, we evaluate the performance of the algorithm using SSFNet [1, 8] simulations framework. Three different variations of trust based on the intermediate nodes are calculated. They are : (i) All nodes are trusted as in a p2p environment (p2p path), (ii) Any Hierarchy node (Any H node), and (iii) Common Ancestors Hierarchy node (CA *H node*). In *p2p path*, any physically reachable neighbor is a trusted intermediate node without any restrictions, as in plain Floyd-Warshall algorithm. In Any H node, a trusted intermediate node is any node higher up in the hierarchy graph, which is randomly generated as part of the input to Algorithm BestOoPPaths. Any nodes higher up in the administrative hierarchy graph H can be trusted in this case. In CA H node, a trusted intermediate node is a common ancestor of source and destination in the hierarchy graph. CA H node is the most restricted form of trust and allows fewer nodes to be included as intermediate nodes by definition when compared to the other two trust mechanisms. A trust matrix is defined for each pair of nodes to hold all of its hierarchical or common ancestors as trusted nodes depending on the algorithm scenario. The hierarchy tree is scanned to fill the entries in parent matrix by a modified in-order traversal.

The simulation framework allowed us to generate realistic graph topologies, to execute QoP algorithms (compiled in), and is currently being also employed to study network traffic performance based on QoP paths. We are also extending the current algorithms with distributed/online versions, for which simulator is ideally suitable.





4.1 Experimental Set Up

In order to generate fair results, we use the BRITE topology generator [18] with the same parameters for the algorithms. N nodes are randomly distributed on a 2-dimensional plane using the Waxman model [25]. The value of the Waxman-specific parameters, α and β were set to 0.15 and 0.2 respectively. We generate QoP values for each edge (in the normalized range 1-10) to represent quality on each edge. The links among all routers are all 10 Mb, while the links connecting routers to end hosts are all 1Mb in capacity. We use SSFNet tools for simulating in detail the structure of realistic networks. In our case, the network is simulated as a single OSPF area. The average time gap between the successive requests is 2 seconds.

We evaluate the average percentage improvement in QoP. This metric refers to the average improvement in the QoP (weight) value obtained from the algorithms over the average initial QoP for all pairs of nodes. The given results are evaluated for 'secure routing' scenario, wherein only trusted nodes are employed in routing.

4.2 Results

In Figure 2, we show the overall percentage improvement in QoP value obtained for all three different scenarios based on varying number of nodes. Note that the p2p path percentage improvement in QoP is the optimum value as it trusts all the intermediate nodes. However, in the other two variations - Any hierarchy node (Any H node) and any common ancestor node (CA H node), performance depends on the position of source-destination pair in the hierarchy graph. Any H node has higher percentage improvement in QoP than CA H node as there is a larger user set with better resource capabilities. The *p2p path* results do not depend on the hierarchy graph at all. Notice that a possible increase in QoP also depends on the placement of hierarchical nodes as adjacent nodes in the original graph. The hierarchy graph H referred in Figure 2 is a spanning tree obtained from the subgraph of initial graph G with 90% edge density. It is noted that for hierarchy graphs generated with lower edge densities from original graph, the percentage improvement in QoP is lower. This is because of fewer possible paths due to increased restriction on intermediate nodes. Our preliminary results for 'secure computing' show that there is increase in percentage improvement for *p2p path* in QoP (around 2% - 8%) when compared to 'secure routing' for the same set of data as in Figure 2 due to inclusion of optimal sub solutions from extended connectivity.

We also measure the topology creation time, system execution time spent by SSFNet in creating the global topology of the network graph including link status announcements, link state database, and the routing tables. This measure represents the system overhead time spent in creating the topology of the network (compile time measure). Hence, it is appropriate to report the system configuration. We ran the experiments for the three scenarios on the same 2.0 GHz Intel Pentium M processor with 786 MB memory and one hard disk with 7200 rpm running Windows XP operating system. The topology execution time ranges for all three variations of algorithms from 0.5-6 seconds for nodes in range 10-1000. There is no significant time difference as building and accessing data structure for trust matrix is the only different operation between p2p and Any H node and CA H node.

5. Related Work

The related work spans into areas of routing, peer nodes for security services, and communication policies.

Routing: Maximum QoP path problem is analogous to maximum bandwidth problem in QoS routing, referred to as widest path. The maximum QoP is computed at an application level. Traditionally, the QoS solution to widest path (maximum bandwidth) is the shortest path algorithms obtained as spanning tree between source and destination pairs [19]. In QoS, routing subject to multiple constraints is NP-complete. Our QoP computation finds the maximum OoP based on a dynamic programming approach subject to trust. The solution required the survey of many shortest path routing algorithms [6], and QoS-routing algorithms. In [17], authors propose a maximum bandwidth path based on Kruskal's minimum weighted spanning trees. The output path obtained from our proposed algorithm corresponds to the best optimization of the objective function available at that instance.

Peers for security services: The idea of using peer nodes

for security services is applied for the certificate chain discovery for authentication purposes [15]. The authors in [15] represent each certificate authority certificate with a coded certificate path label and design an algorithm to speed up the process of certificate path discovery in an infrastructure-less environment with the aid of other nodes. We formulate the problem in general to address other security services and detail on the encryption service in the paper.

Communication policies: The appropriate policy specification for communication depends on kind and size of collaborative group. In [14], the member policies are applied to negotiate security policy for communications within coalitions. Multidimensional Security Management and Enforcement (MSME) binds the abstract requirements from policies to the service mechanisms which can be enforced at different levels of the TCP/IP stack. For example, suppose that there is a defined service 'enc' for encryption, specified in the abstract form as (enc, type, key length), where 'type' is one of 'symmetric block', 'symmetric stream' or 'public key', and key length specifies a minimum key length in bits. The encryption part of its rule actions takes the form (cipher type, length), where cipher type is the name of an encryption algorithm and length is a key length in bits specified either as an exact number or as a range [a-b]. In our approach, we extend the pair-wise communication to all-to-all pairs, expanding communication network beyond pair-wise intersections. Our communication policy representation is embedded into a graph representation.

6. Conclusions

We introduced the communication coalition where the members work collectively to improve the QoP along the communication paths. The communication messages are directed across those paths where the medium is more secure or where there are more safeguards in terms of implementation resources for encryption and other security measures. Our graph based approach for communication increases the overall connectivity and improves QoP of the coalition members. Our results also indicate a significant increase in the average QoP value. The members who are not able to communicate earlier can communicate based on the trusted peer members as intermediaries.

Our approach of using QoP metric based on encryption strengths builds secure communication path and improves the coalition communication. We look into developing algorithms and simulations to extend the proposed idea for distributed and dynamic routing approach of the problem. In future, we would like to see how other security services, such as availability, confidentiality, integrity, etc., are affected from the intermediary nodes and assess using empirical quantifications. Integrating routing constraints in QoS and security is another interesting area to be considered. Routing mechanisms based on different hardware capabilities can also be explored. We plan to extend the p2p path between any two pairs to a sub-group of arbitrary size.

References

- [1] http://www.ssfnet.org, last accessed on July 02, 2006.
- [2] Web-services security quality of protection. In *Whitepaper, Entrust technologies*, 2002.
- [3] A short tutorial on distributed pki. In *White paper, Isode*, 2006.
- [4] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.
- [5] I. A. Almerhag and M. E. Woodward. Security as a quality of service routing problem. In *CoNEXT: ACM conference on Emerging network experiment and technology*, 2005.
- [6] B. Awerbuch. Shortest paths and loop-free routing in dynamic networks. In Proc. of the Annual ACM SIGCOMM Symposium on Communication Architectures and Protocols, 1990.
- [7] E. Charles, J. Phillips, T. C. Ting, and S. A. Demurjian. Information sharing and security in dynamic coalitions. In *Proc. of the 7th ACM symposium on Access control models and technologies*, 2002.
- [8] J. Cowie, H. Liu, J. Liu, D. Nicol, and A. Ogielski. Towards realistic million-node internet simulations. In *Proc. of the 1999 Intnl. Conf. on Parallel and Distributed Processing Techniques and Applications*, 1999.
- [9] W. Dai. http://www.eskimo.com/ weidai/benchmarks.html.
- [10] P. T. Dinsmore, D. M. Balenson, H. M. Kruus, P. S. Scace, and A. T. Sherman. Policy-based security management for large dynamicgroups: An overview of the DCCM project. In *Proc. of the DARPA Information Survivability Conf. & Exposition Volume I of II (DISCEX)*, pages 64–73, 2000.
 [11] Y. Duan, J. Wang, M. Kam, and J. Canny. Privacy preserving
- [11] Y. Duan, J. Wang, M. Kam, and J. Canny. Privacy preserving link analysis on dynamic weighted graph. *Comput. Math. Organ. Theory*, 11(2), 2005.
- [12] L. Elbaz and H. Bar-El. Strength assessment of encryption algorithms. White paper, 2000.
- [13] S. Foley, S. Bistaelli, B. OSullivan, J. Herbert, and G. Swart. Multilevel security and quality of protection. In *First Workshop on Quality of Protection*, 2005.
- [14] P. G.Condell, M. R. Krishnan, and L. Sanchez. Multidimensional security policy management for dynamic coalitions. In *DISCEX*, 2001.
- [15] H. Huang and S. F. Wu. An approach to certificate path discovery in mobile ad hoc networks. In *Proc. of the 1st ACM wkshp on Security of ad hoc and sensor networks*, 2003.
 [16] S. Malladi, S. K. Prasad, and S. B. Navathe. An im-
- [16] S. Malladi, S. K. Prasad, and S. B. Navathe. An improved quality of protection for secure communication policy agreements. Technical report CS-TR-05-04, Georgia State University, Department of Computer Science, 2005. http://suez.cs.gsu.edu/cscsrmx/mediatortechreport.pdf.
- [17] N. Malpani and J. Chen. A note on practical construction of maximum bandwidth paths. volume 83, 2002.
- [18] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: An approach to universal topology generation. In In Proc. of the Intnl. Wkshp. on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2001.
- [19] P. Paul and S. V. Raghavan. Survey of QoS routing. In Proc. of the 15th Intl. conf. on Comp. comm., pages 50–75, 2002.
- [20] B. Potter and B. Fleck. 802.11 Security. O'Reilly & Associates, Inc., 2003.

- [21] M. Satyanarayanan. Pervasive computing: Vision and challenges. In *IEEE Personal Communications*, pages 10–17, aug 2001.
- [22] G. Suryanarayana, M. H. Diallo, J. R. Erenkrantz, and R. N. Taylor. Architectural support for trust models in decentralized applications. In *ICSE '06: Procs. of the 28th Intl. conf. on Software engineering*, pages 52–61, 2006.
- [23] Tedijanto, T. E. Onvural, R. Verma, L. Gun, and R. A. Guerin. NBBS path selection framework - networking broadband services, 1995.
- [24] U. Varshney. Wireless networks for enhanced monitoring of patients. In *International Journal of Healthcare Technology* and Management, volume 6, pages 489 – 499, 2005.
- [25] B. M. Waxman. Routing of multipoint connections. In *IEEE Journal on Selected Areas in Communication*, volume 6, pages 1617–1622, 1988.
- [26] D. Wijesekera and S. Jajodia. Policy algebras for access control: the propositional case. In *Proc. of the 8th ACM conference on Computer and Communications Security*, 2001.
- [27] T. Xie, X. Qin, and A. Sung. SAREC: A security-aware scheduling strategy for real-time applications on clusters. In *ICPP '05: Procs. of the 2005 Intl. Conf. on Parallel Processing*, pages 5–12, 2005.