Linear Iterations on Ordered Semirings for Trust Metric Computation and Attack Resiliency Evaluation

George Theodorakopoulos and John S. Baras
Institute for System Research
Department of Electrical and Computer Engineering
University of Maryland
College Park, Maryland 20742
Email: {gtheodor,baras}@isr.umd.edu

***Abstract*— Within the realm of network security, we interpret the concept of trust as a relation among entities that participate in various protocols. Trust relations are based on evidence created by the previous interactions of entities within a protocol. In this work, we are focusing on the evaluation of trust evidence in Ad Hoc Networks. Because of the dynamic nature of Ad Hoc Networks, trust evidence may be uncertain and incomplete. Also, no pre-established infrastructure can be assumed. The evaluation process is modelled as a path problem on a directed graph, where nodes represent entities, and edges represent trust relations. We develop a novel formulation of trust computation as linear iterations on ordered semirings. Using the theory of semirings, we analyze several key problems on the performance of trust algorithms. We also analyze the resilience to attacks of the resulting schemes.**

I. Trust - Definition - Motivation

The notion of trust, in the realm of network security, will for our purposes correspond to a set of relations among entities that participate in a protocol. These relations are based on the evidence generated by the previous interactions of entities within a protocol. In general, if the interactions have been faithful to the protocol, then trust will "accumulate" between these entities. Exactly how trust is computed depends on the particular protocol (application). The application determines the exact semantics of trust, and the entity determines how the trust relation will be used in the ensuing steps of the protocol. Trust influences decisions like access control, choice of public keys, etc. It could be useful as a complement to a Public Key Infrastructure (PKI), where an entity would accept or reject a public key according to the trustworthiness of the entities that vouch for it (i.e. have signed a certificate for it) – this is the idea behind PGP's Web of Trust [3]. It can also be used for routing decisions: Instead of the shortest path, we could be looking for the most trusted path between two nodes (this has been already proposed in P2P networks [4]).

In this work we model and analyze trust schemes for mobile ad hoc networks (MANET). Ad Hoc networks are envisioned to have dynamic, sometimes rapidly-changing, random, multihop topologies which are composed of bandwidth-constrained wireless links. The nodes themselves form the network routing infrastructure in an ad hoc fashion [5]. Based on these characteristics, we are imposing the following constraints on our schemes:

First, there is no preestablished infrastructure. The computation process cannot rely on, e.g., a Trusted Third Party. There is no centralized Public Key Infrastructure, Certification Authorities, or Registration Authorities with elevated privileges.

Second, evidence is uncertain and incomplete. Uncertain, because it is generated by the users on the fly, without lengthy processes. Incomplete, because in the presence of adversaries we cannot assume that all friendly nodes will be reachable: the malicious users may have rendered a small or big part of the network unreachable. Despite the above, we require that the results are as accurate as possible, yet robust in the presence of attackers. It is desirable to, for instance, identify all allied nodes, but it is even more desirable that no adversary is misidentified as good.

In this work we do not assume the existence of any globally trusted entity: on the contrary, everything is up to the individual nodes of the network. They themselves sign certificates for each other's keys, and they themselves have to judge how much to trust these certificates and, essentially, their issuers.

The specification of admissible types of evidence, the generation, distribution, discovery and evaluation of trust evidence are collectively called Trust Establishment. In this work, we are focusing on the evaluation process of trust evidence in Ad-Hoc Networks, i.e. we are focusing on the trust metric itself. In particular, we are not dealing with the collection of evidence from the network, and the accompanying communication and signaling overhead. This issue is important, and obviously needs to be addressed in a complete system.

Trust computation is the application of a metric to a body of evidence. This evidence is based on interactions of users within a network, and the result of the computation

("trust") is a quantitative belief of User i about User j's behavior (i.e. is User j trustworthy according to User i?).

For example, assume we have a wireless network where users are supposed to forward data they receive. An interaction in this setting would play out as follows: User i sends a packet to his neighbor. The neighbor has the choice to either forward the packet (as he is supposed to), or drop it (since he may not want to waste his energy). This choice is observed by User i, and counts either as a Good or as a Bad interaction, respectively. Repeated interactions of this type build up the previously mentioned evidence, which will be called *direct opinions*.

The trust computation will compute *indirect opinions*, that is, opinions of a user for others with which he has had no previous direct interaction. The idea is to take advantage of the interactions (and thus the direct opinions) that intermediate users have had with each other.

II. Trust as a path problem

We treat the trust computation problem as a generalized shortest path problem on a weighted directed graph $G(V, E)$ (*trust graph*). The vertices of the graph are the users/entities in the network. A weighted edge from vertex i to vertex j corresponds to the *opinion* that entity i has about entity j. The weight function is $w(i, j) : V \times V \longrightarrow S$, where S is the opinion space. The set S and the precise semantics of opinions are parameters of the model and can differ according to the application.

Assume that User s wants to compute the trustworthiness of User d. So, s will ask the people he knows (i.e. has an opinion about) and they will tell him their opinion about d, or they will ask the users they know, etc., until persons with a direct interaction with d are found. Formally, all the direct information that exists about the destination d, is contained in the weighted, directed edges that point to d. On the other hand, all the direct information that s has about the rest of the network is contained in s's outgoing edges. In effect, s knows about the rest of the network only through his one-hop trust neighbors. Therefore, *all information about d that s can use is contained in the paths from s to d*. For example, edges pointing to d (or, in general, directed paths to d) that are not reachable from s are useless, and edges pointing out from s that are not on directed paths from s to d are dead ends. This is the starting observation of this work, and the reason why the subsequent model was chosen: because it fits the path-based nature of the problem.

Along each path, concatenation of opinions occurs: If s has opinion w_{s1} about 1, and 1 has opinion w_{12} about 2, then s can form an indirect opinion w_{s2} about 2 that is a function of w_{s1} and w_{12} (denoted by $w_{s1} \otimes w_{12}$). If there are multiple paths from s to d, then indirect opinions from each path are aggregated to form the overall opinion of s for d (denoted by $t_{sd} = t_{sd}^{p_1} \oplus t_{sd}^{p_2} \oplus \cdots \oplus t_{sd}^{p_n}$, where the p_i's are the paths from s to d.).

We have mentioned two operations: one is the *concatenation* of opinions along a path and the other is the *aggregation* across paths.

$$T_{s,1} \geq T_{s,2} \geq T_{s,3} \geq T_{s,d}$$

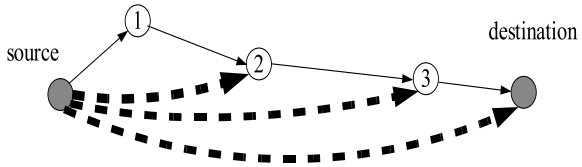

Fig. 1. Concatenation of opinions

$$T_{s,d} \geq T_1$$
$$T_{s,d} \geq T_2$$

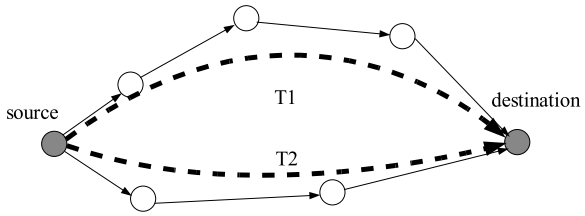

Fig. 2. Aggregation of opinions

These operators, along with the carrier set S, form a semiring $(S, \oplus, \otimes)$:

- $\oplus$ is commutative, associative, with a neutral element $⓪ \in S$.
- $\otimes$ is associative, with a neutral element $① \in S$, and $⓪$ as an absorbing element.
- $\otimes$ distributes over $\oplus$.

In addition, the ordering relations described in Figures 1 and 2, regarding concatenation of trust along a path and aggregation of trust across paths, above, introduce a partial order over our semiring, and thus the semiring we are considering is *an ordered semiring* [6].

The semiring property is very desirable because it fits the path-based nature of the problem: many other path problems can be expressed as semiring computations [6]. For example, suppose the edge weights are transmission delays, and we want to compute the least delay path from i to j. The semiring to use is $(\Re_+ \cup \{\infty\}, \min, +)$, i.e. $\oplus$ is $\min$, and $\otimes$ is $+$: The total delay of a path is equal to the sum of all constituent edge delays, whereas the shortest path is the one with minimum delay among all paths. Also, $⓪$ is ∞, and $①$ is 0. On the other hand, if edge weights are link capacities, then the maximum bottleneck capacity path is found by the semiring $(\Re_+ \cup \{\infty\}, \max, \min)$, with

⓪ $\equiv 0, ① \equiv \infty$. Then, the result d_{ij} is equal to the maximum rate of traffic that i can send to j along a single link. The transitive closure of a graph uses the Boolean semiring: $(\{0,1\}, \vee, \wedge)$, where all edge weights are equal to 1. This answers the problem of path existence from i to j, i.e. $d_{ij} = 1$ if and only if there exists an $i \longrightarrow j$ path.

We now look at the expected trust behavior of the operators.

- First, we don't want B to be able to increase A's trust in C beyond A's trust in B. For instance, assume A trusts B only moderately, and B trusts C a lot. Then it makes sense that A's trust in C is also moderate, since A has only B's word to count on. In general, concatenation should not increase trust. Note that the total opinion along a path is "limited" by the source's opinion for the first node in the path.
- Second, it is better to have multiple independent opinion paths to the destination. In principle, the more independent information there is, the better decision the source can reach. For example, path independence in Public Key Authentication has been argued in [2]. In order to quantify this case, we require the aggregation operator to increase something about the resulting opinion. However, trust cannot increase. If, say, there are multiple opinions all saying that the destination is untrustworthy, then obviously the source's aggregate opinion should be along these lines. So, for our own semiring we introduce an extra parameter (or better metric) called *confidence* which is what increases when we have multiple paths.

Thus the setting and formalism we have introduced for trust computation is more along the lines of multicriteria (or multi metrics) computation within an ordered semiring [6].

A. Our Semiring

Each opinion consists of two numbers: the *trust* value, and the *confidence* value. The former corresponds to the issuer's estimate of the target's trustworthiness. For example, a high trust value may mean that the target is an ally (in a military setting), or that the target has been honest in his past business transactions, or that a digital certificate issued for the target's public key is believed to be correct. On the other hand, the confidence value corresponds to the accuracy of the trust value assignment. A high confidence value means that the target has passed a large number of tests that the issuer has set, or that the issuer has interacted with the target for a long time, and no evidence for malicious behavior has appeared. Since opinions with a high confidence value are more useful in making trust decisions, the confidence value is also referred to as the *quality* of the opinion.

In our semiring, the opinion space is $S = [0,1] \times [0,1]$

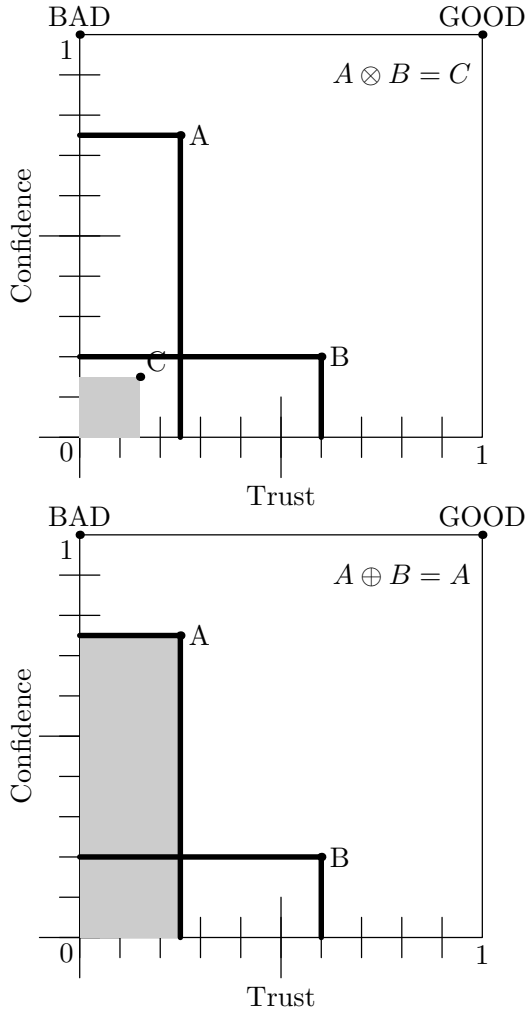

Fig. 3. $\otimes$ and $\oplus$ operators

Our choice for the $\otimes$ and $\oplus$ operators is as follows (Fig. 3):

$$(t_{ik}, c_{ik}) \otimes (t_{kj}, c_{kj}) = (t_{ik}t_{kj}, c_{ik}c_{kj}) \quad (1)$$

$$(t_{ij}^{p_1}, c_{ij}^{p_1}) \oplus (t_{ij}^{p_2}, c_{ij}^{p_2}) = \begin{cases} (t_{ij}^{p_1}, c_{ij}^{p_1}) & \text{if } c_{ij}^{p_1} > c_{ij}^{p_2} \\ (t_{ij}^{p_2}, c_{ij}^{p_2}) & \text{if } c_{ij}^{p_1} < c_{ij}^{p_2} \\ (t_{ij}^{*}, c_{ij}^{p_1}) & \text{if } c_{ij}^{p_1} = c_{ij}^{p_2} \end{cases} \quad (2)$$

where $(t_{ij}^{p_1}, c_{ij}^{p_1})$ is the opinion that i has formed about j along the path p_1, and $t_{ij}^{*} = \max(t_{ij}^{p_1}, t_{ij}^{p_2})$.

We can verify by direct substitution that the neutral elements for this semiring are: $⓪ = (t, 0)$, for any t, and $① = (1, 1)$.

III. Trust as a system of equations - Attacker Game

We have seen that what we want to compute is the following semiring-summation over all the paths p from s to d.

$$t_{sd} = \bigoplus_{p} t_{sd}^{p}$$

We can break up these paths according to their last link, and so we get:

$$t_{sd} = \bigoplus_{k \in N_d} t_{sk} \otimes w(k,d)$$

where N_d are the in-neighbors of d: the users that have a direct opinion about d. If we now let d vary over the set of all users, t_{sd} becomes a vector, and we can write a vector equation:

$$\vec{t} = \vec{t}W \tag{3}$$

where W is the matrix of direct opinions. So, the result of the trust computation for User s (s's indirect opinions about everybody else), is the eigenvector of W associated with ① (the neutral element for $\otimes$, which in our semiring is also the maximum element).

It is natural to expect that each user will have the maximum direct opinion about himself, i.e. $\forall i, w(i,i) = (1,1) = ①$. Notice that the vector $\vec{t}$ also includes the indirect opinion of User s about himself, which we intuitively expect to be ① . This is guaranteed through setting $w(i,i) = ①$.

So, we have formulated the problem of computing indirect opinions as an eigenvector problem. Perron-Frobenius theory for semirings (see e.g. Baccelli, Cohen, Olsder, and Quadrat [1, Thm 3.23]) tells us that if W is irreducible (i.e. the graph is strongly connected, which we assume here), then there exists exactly one eigenvalue, but possibly many eigenvectors. This eigenvalue λ is equal to the maximum mean circuit weight of the graph. Using semiring operators this is written as

$$\lambda = \bigoplus_{j=1}^{n} (\mathrm{trace}(A^j))^{\frac{1}{j}}$$

and in our semiring it would correspond to the geometric mean of a path's weights.

Using Theorem 3.101 from [1]:

Theorem 1: Given a matrix A with maximum circuit weight ① , any eigenvector associated with the eigenvalue ① is obtained by a linear combination of N_A^c columns of A^+, where N_A^c denotes the number of strongly connected components of $G^c(A)$ (the critical graph of A, equal to the union of the critical circuits). In particular, each column corresponding to a node in the critical graph is an eigenvector, and columns corresponding to nodes in separate components are "linearly independent".

we have a characterization of the set of eigenvectors associated with the eigenvalue ① . The critical circuits are the circuits of W that have maximum mean weight (equal to λ), and the matrix $W^+ = W \oplus W^2 \oplus \ldots$. Since $W_{ii} = w(i,i) = ①$, and ① is the maximum opinion value, the self loops $i \rightsquigarrow i$ are critical circuits. Other critical circuits would have to consist exclusively of direct opinions equal to ① , i.e. a cycle of users *completely* trusting each other, which we assume is not the case. The union of the critical circuits is a graph with one strongly connected component per user, so using the theorem above we see that there is one independent eigenvector per user.

These eigenvectors all have eigenvalue equal to ① , so in principle they could all fit (3). But for each s, there is only one eigenvector that gives an indirect opinion about s equal to ① . That is the eigenvector equal to row s of the matrix W^+. It is a row and not a column of W^+, because we defined $W_{ij} = w(i,j)$ to be equal to the direct opinion of i about j. In [1], A_{ij} is the weight of the edge $j \rightarrow i$.

The matrix W^+ is in general an infinite sum, so there are issues of convergence. In our case, convergence is not only guaranteed, but it happens in a finite number of steps. The reason is that there are no circuits with weights (see also [1, Thm 3.20]) greater than ① , so by including cycles in our path computations, we do not increase the computed trust values. This situation is identical to a shortest path problem with no cycles of negative length.

The number of steps required for convergence is at most equal to the total number of users in the network. This is the case because the partial sum up to the k-th power gives the maximum path weights from i to j among the k-link paths. Since no maximum path contains a cycle (except the trivial self-loops), no maximum path can have more links than the number of users. This is independent of the topology, assuming, of course, that the graph is strongly connected.

Suppose now that there exists an Attacker who wants to manipulate the trust computation, i.e. cause User s to compute false opinions about others. The Attacker can change the opinion on a single edge, which would amount to tricking a user into issuing a false opinion, or creating a forged opinion. We want to see what is the maximum damage the Attacker can cause. This is equivalent to asking what single entry change in the matrix W causes the largest change in the eigenvector. The Attacker causes W to become W^*, so t becomes t^*. The damage is equal to $||t - t^*||$, where $||\cdot||$ is a suitable (e.g. the L_1 or the L_∞) norm.

In what follows we limit our attention to a particular pair s-d. We will examine which edge the Attacker will attack, and characterize the resilience of the s-d trust computation to such single edge attacks.

The problem just described is very similar to computation of tolerances for edges of a network. In short, if p^* is an optimal path from s to d, the upper (lower) tolerance of an edge e with respect to p^* is the largest (smallest) weight of edge e that preserves the optimality of p^*. The most vital edge is defined as the edge that, if deleted, causes the greatest deterioration in the optimal path weight. Our main reference is the work by Ramaswamy, Orlin, and Chakravarti [13]. We will describe their results, and then unify and generalize them.

For a concrete example, we will use (as in [13]) the shortest path problem in an undirected graph $G = (V, E)$ with nonnegative weights $c_{ij} \geq 0, (i,j) \in E$. Let p^* be a shortest path from s to d, $s, d \in V$, and upper/lower tolerances also defined as above. Intuitively, an Attacker

that wants to increase the shortest path weight will necessarily delete an edge on p^*. Deleting is always worse than merely increasing the weight of the edge, and deleting an edge off the path has no effect on the optimal weight. But it makes a difference which of the edges the Attacker will delete, since the respective new shortest paths will in general differ. The worst edge is the one that is, in a sense, "harder" to replace; hence the name "most vital edge". The most vital edge turns out to be the edge with the largest upper tolerance.

The results of [13] for the tolerances of edges in the shortest path problem are summarized below (Corollary 1 in [13]), where α_e is the lower tolerance of edge e, and β_e the upper, $d^{e\leftarrow x}(s,d)$ is the shortest $s \rightsquigarrow d$ path weight when the weight of edge e is x, and $c(p^*), c_e$ are the weights of the path p^* and edge e, respectively.

Theorem 2 (Shortest Path Tolerances): Let p^* be a shortest path in $G = (V, E)$.

1) If $e \in p^*$, then
   - $\alpha_e = 0$, and
   - $\beta_e = d^{e\leftarrow\infty}(s,d) - c(p^*) + c_e$.
2) If $e \notin p^*$, then
   - $\alpha_e = c(p^*) - d^{e\leftarrow 0}(s,d)$, and
   - $\beta_e = \infty$.

In the same paper, the authors also address tolerances in the context of, as they name it, problems with bottleneck objectives. In these problems, as we have discussed in Section II, the edge weights represent link capacities, and we are looking for an $s \rightsquigarrow d$ path whose bottleneck (minimum capacity link) is the maximum. Their results are summarized below (Corollary 2 in [13]), where $G^{e\leftarrow x}$ is the graph G with the weight of edge e changed to x, $v(G)$ is the maximum capacity path weight in G, and $c(p^*)$ is now the bottleneck capacity of p^*.

Theorem 3 (Maximum Capacity Tolerances): Let p^* be a maximum capacity path in $G = (V, E)$.

1) If $e \in p^*$, then
   - $\alpha_e = v(G^{e\leftarrow -\infty})$.
   - If p^* is a maximum capacity path in $G^{e\leftarrow\infty}$, then $\beta_e = \infty$.
   - If p^* is not a maximum capacity path in $G^{e\leftarrow\infty}$, then β_e is the minimum capacity of an edge of $p^* \setminus e$.
2) If $e \notin p^*$, then
   - $\alpha_e = -\infty$.
   - If p^* is a maximum capacity path in $G^{e\leftarrow\infty}$, then $\beta_e = \infty$.
   - If p^* is not a maximum capacity path in $G^{e\leftarrow\infty}$, then $\beta_e = c(p^*)$

The authors treat these problems as different, but we will now show that they can be described within the same framework, as soon as we realize that they are both semiring path problems. Our main result is the following:

Theorem 4 (Semiring Tolerances): Let OPT^* be the set of $\oplus$-optimal paths in $G = (V, E)$. Instead of lower and upper tolerances, α_e and β_e now mean $\oplus$-minimal and $\oplus$-maximal values of an edge $e \in E$ that preserve the $\oplus$-optimality of some path in OPT^*.

1) If $\exists p^* \in OPT^* : e \in p^*$, then
   - $\alpha_e = \left(\bigoplus_{\substack{p:s\rightsquigarrow d \\ w(e)\leftarrow \mathbb{0}}} w(p) \right) \oslash w(p^* \setminus e)$. Moreover, if $\exists p^* \in OPT^* : e \notin p^*$, then $\alpha_e = \mathbb{0}$.
   - $\beta_e = \mathbb{1}$.
2) If $\nexists p^* \in OPT^* : e \in p^*$, then
   - $\alpha_e = \mathbb{0}$. It suffices that $\exists p^* \in OPT^* : e \notin p^*$.
   - $\beta_e = w(p^*) \oslash \left(\bigoplus_{\substack{p:s\rightsquigarrow d \\ w(e)\leftarrow \mathbb{1}}} w(p) \right)$

The operator $\oslash$ is the inverse of $\otimes$. Since we are dealing with semirings, $\oslash$ may not always be defined, as in the case of $\otimes = \min$. In these cases, $a = b \oslash c$ means that a, b, and c are such that the equality $a \otimes c = b$ holds. We can verify by substitution that Theorem 4 holds for the two specific problems mentioned above.

The benefit of this generalization is that we can directly apply it to semirings where $\oplus$ is $\max$ or $\min$, i.e. where there is some optimization involved. Our trust semiring is $(\oplus, \otimes, \mathbb{0}, \mathbb{1}) = (\max, \cdot, 0, 1)$, so we can directly apply Theorem 4. Lower tolerance is α_e, upper tolerance is β_e.

1) If $\exists p^* \in OPT^* : e \in p^*$, then
   - $\alpha_e = \frac{w(e)}{w(p^*)} \cdot \left(\max_{w(e)\leftarrow 0} w(p)\right)$. Moreover, if $\exists p^* \in OPT^* : e \notin p^*$, then $\alpha_e = 0$.
   - $\beta_e = 1$.
2) If $\nexists p^* \in OPT^* : e \in p^*$, then
   - $\alpha_e = 0$. It suffices that $\exists p^* \in OPT^* : e \notin p^*$.
   - $\beta_e = \dfrac{w(p^*)}{\max_{w(e)\leftarrow 1} w(p)}$

If the user d, for which s is computing the indirect opinion, is a good user, then the Attacker will want to reduce the computed opinion. In that case, the link to be attacked is the one with the smallest lower tolerance α_e. The attack will consist of setting the weight of the edge at 0. If, on the other hand, d is a bad user, then the Attacker will try to increase the computed indirect opinion. So, he will attack the edge with the largest upper tolerance, and set its weight to 1.

The mathematical techniques we use come from extensions of Perron Frobenius theory over semirings and ordered semirings, eigenvectors of monotone functions, and idempotent semirings, for which the interested reader can consult the following references [6], [7], [8], [9], [10], [11].

IV. Conclusion

We have presented a trust computation framework, and linked its properties to properties of mobile ad-hoc networks. The mathematical foundation for the computation is the theory of semirings and matrix iterations over them. We have shown what the canonical issues of the theory

(eigenvectors, convergence, speed of convergence) mean for our application. We have also generalized previous work on edge tolerance computation and used it to compute the attack resilience of the trust computation.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Army Research Office under CIP URI grant No DAAD 19-01-1-0494. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U.S. Army Research Office.

REFERENCES

[1] F. L. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat, *Synchronization and Linearity: An Algebra for Discrete Event Systems*, John Wiley & Sons, 1992.

[2] M. K. Reiter and S. G. Stubblebine, Resilient Authentication Using Path Independence, *IEEE Trans. Comput.*, 47(12):1351–1362, December 1998.

[3] P. R. Zimmermann, *The Official PGP User's Guide*, MIT Press, 1995

[4] S. Marti, P. Ganesan and H. Garcia-Molina, *SPROUT: P2P Routing with Social Networks*, Stanford University, 2004.

[5] S. Corson and J. Macker, Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501, IETF, January, 1999.

[6] G. Rote, Path problems in graphs, in G. Tinhofer, E. Mayr, H. Noltemeier, and M. M. Syslo, editors, *Computational Graphs Theory*, volume 7 of Computing Supplementum, Springer-Verlag, 1990.

[7] D. de Falco, M. Goldwurm, V. Lonati, Frequency of Symbol Occurrences in Bicomponent Stochastic Models, *INRIA Report*, 2003.

[8] A. Bertoni, C. Choffrut, M. Goldwurm, V. Lonati, Local Limit Distributions in Pattern Statistics: Beyond the Markovian Models, Rapport de Recherche L.I.A.F.A. n. 2003-019 Laboratoire d' Informatique Algorithmique, Fondements et Applications Universite Paris VII, 2 Place Jussieu, 75221 Paris, 2003.

[9] L. B. Beasley, A. E. Guterman, S. G. Leey, S. Z. Song, Determinant Preservers for Matrices over Semirings, *LIN. ALG. & APPL.* 2003.

[10] S. Gaubert and J. Gunawardena Existence of Eigenvectors for Monotone Homogeneous Functions, *Technical Report HPL-BRIMS-1999-08*, Basic Research Institute in the Mathematical Sciences, HP Laboratories Bristol, 1999.

[11] J. Gunawardena, Editor, *Idempotency*, Cambridge University Press, 1998.

[12] R. E. Tarjan, Sensitivity Analysis of Minimum Spanning Trees and Shortest Path Trees. INFO. PROC. LETT. Vol. 14, no. 1, pp. 30-33. 1982

[13] R. Ramaswamy, J. B. Orlin, and N. Chakravarti, Sensitivity Analysis for Shortest Path Problems and Maximum Capacity Path Problems in Undirected Graphs. June 2003. MIT Sloan Working Paper No. 4465-03. Available at SSRN: http://ssrn.com/abstract=489804