Attack-Resistance of Computational Trust Models

Andrew Twigg and Nathan Dimmock
Computer Laboratory, Cambridge University, UK

E-mail: firstname.lastname@cl.cam.ac.uk

Abstract

The World Wide Web encourages widely-distributed, open, decentralised systems that span multiple administrative domains. Recent research has turned to trust management *[4] as a framework for decentralising security decisions in such systems. However, whilst traditional security measures such as cryptography and encryption are well-understood (theoretically and empirically), the same cannot be said for computational trust models. This paper describes the attack-resistance of several well-referenced trust models, in a move toward a possible framework and terminology for such analyses. We present a number of open questions, and consider possible future directions in the area.*

1 Why Computational Trust Models?

The World Wide Web encourages widely-distributed, open systems that span multiple administrative domains. Unfortunately, the characteristics of such systems mean that one cannot rely solely on traditional security measures. These 'open distributed systems' have a number of characteristics:

- relationships are on a peer-to-peer basis;
- many peers will have never previously interacted;
- multiple administrative domains;
- the lack of any *globally trusted* third party.

The concept of *trust management* [4] provides a framework for decentralising security decisions, which appears able to provide a different 'paradigm of security' in such systems. However, whilst traditional security measures such as cryptography and encryption are well-understood (theoretically and empirically), the same cannot be said for computational trust models. We start by sampling a number of well-referenced trust metrics, then present some terminology for assessing and reasoning about their attack-resistance.

2 A Sampling of Trust Metrics

At its heart, a computational *trust model* contains a *trust metric*. Reiter and Stubblebine [12] consider the problem of authenticating entities using public-key certification in a large-scale, open, distributed system with no trusted third party to manage the name-key bindings of entities. In this context, a trust model takes as input a set of certificates between keys, a *source* node and a *target* node, where the source wishes to determine the name-key binding for the target. A *trust metric* operates over a *certification graph* that encodes the trust (certificate) relationships between keys, and returns a *trust value* which represents how trustworthy the source deems the target name-key binding to be.

The problem is this: an *attacker* wishes to introduce a false name-key binding (to impersonate another entity), known as a *forgery*. The goal of the trust metric is to resist such attacks by rejecting the forgery. The remainder of this section is devoted to a sampling of trust metrics, concentrating on their attack-resistance.

However, trust metrics have a much wider field of application than avoiding forged name-key bindings. We consider trust metrics which operate in the following sense. There is a directed graph G where nodes represent *principals* and weighted edges represent trust relationships between principles, weighted by a *trust value*. The metric takes a source and a target principle and determines a trust value between them. As an example, consider recommendation-based trust metrics. The graph is a 'recommendation graph' where nodes represent principals and an edge (u, v) with label r means that the current node has a recommendation r from principal u, about principal v. Before investigating trust metrics, we present a brief set of terminology.

An *attack* on a graph G is represented by a new graph G' which contains at least one new target node, known as the *forgery*, *e.g.* a certification graph attack G' on the set of keys V (known as the *victims*) allows new or changed edges only from the victims (corresponding to stealing nodes' secret keys). Figure 1

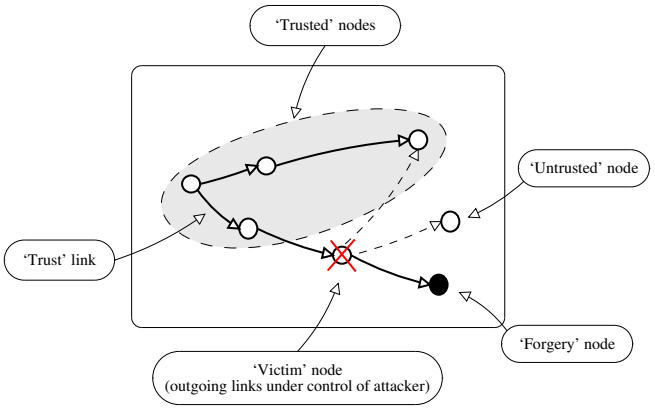

Figure 1. The trust graph representation

shows a trust graph along with this terminology.

We say that an α-attack G' introduces α forgeries into G, and is *successful* if, for each forgery x, the proportion of source nodes which accept x in G' is at least the proportion of source nodes which accepted the target it replaced in G. Let the *cost* of an α-attack on p victims be (p/α), defined in this way because we are interested not in the total cost to the attacker (number of victims), but in the cost per forgery successfully introduced. Then we say a trust metric M is γ-resistant if there is no successful attack on M with cost $< \gamma$. In this paper, we consider worst-case rather than distributional or competitive analysis. All these forms of analysis are justifiable, but the former is the simplest.

2.1 Some Basic Trust Metrics

We start by looking at two trust metrics which represent the simplest non-trivial class of trust metrics, *i.e.* excluding those which blindly accept or reject target nodes.

Reachability. The most basic computational trust metric allows a source to accept (deem as trustworthy) a target node if, and only if, there is a certification path between the source and the target, in the certification graph.

Bounded Reachability (X.509). The bounded reachability trust metric extends the Reachability metric by only allowing certification paths with length less than some specified value, k. This is essentially the 'trust metric' used in the X.509 public-key infrastructure, although X.509 makes use of a small number of certification authorities (CA) (not necessarily trusted third parties) which are supposed to be difficult to compromise.

2.1.1 Attack-resistance

Under our simplifying assumption that all nodes are equally-easily attacked[1], both these trust metrics have low attack-resistance. In the worst-case, an attack needs only a single victim (corresponding to stealing a node's private key) in order to force the metric to accept *any forgery* the attacker wishes. Hence, we say these trust metrics are not γ-resistant for $\gamma > 1$, since a successful attack needs only 1 *victim*.

2.2 Basic Trust Metrics with 'Extra Features'

The class of trust metrics presented here are those which are fundamentally the same as the previous class, yet can appear quite complex (and the majority of those in the literature fall into this class). What we want to know is, does this extra complexity provide extra attack-resistance? Essentially, the answer appears to be 'no'.

We call a trust metric *local* if the values it computes are based on local estimates of trust in the graph. More precisely, the trust value $T_{s,t}$ computed by an ε-local trust metric $M(G)$ on G is altered by at most ε by the removal of a node not on a path from the source s to the target t, in G. Since trust values are often taken to be in $[0, 1]$, metrics which do not rely on recommendations from nodes not on a path to the target are 0-local. Those which use recommendations without discounting them (*e.g.* Beth-Borcherding-Klein) are 1-local.

Aberer-Despotovic [2]. This is a metric based on complaints issued between peers. A target node is considered to be trustworthy if the product of number of complaints received and number issued by that node does not deviate too much from the average such product, over all nodes. Aside from difficulties in estimating this average in a distributed environment, the metric is not γ-resistant for $\gamma > 1$, since an attacker can attack a single victim and make an arbitrary number of complaints about a target node, forcing the target node to be accepted or rejected by the trust metric.

Beth-Borcherding-Klein [3]. The metric computes a trust value based on all the paths $s \rightsquigarrow t$, using

[1]The motivation for this assumption is that we would like to consider trust metrics for fully-decentralized open systems, so the notion of 'secure' and 'trusted' third parties is not appropriate

direct and recommendation trust values.

$$T_{s,t} = 1 - \prod_{i=1}^{m} \sqrt[n_i]{\prod_{j=1}^{n_i}(1 - v_{i,j})} \quad (1)$$

where there are n_i distinct paths with trust values $v_{i,1}, v_{i,2}, \ldots, v_{i,n_i}$. Reiter and Stubblebine [12] show that, using a single victim, an attacker can drive the result of this trust metric *arbitrarily close* to any desired value.

Rahman-Hailes [1]. The metric computes the trust value of a path as a product of the trust values of edges on the path, and takes the average of all path values:

$$T_{s,t} = 1/N \cdot \sum_{i=1}^{N} \left(v_t \cdot \prod_{j=1}^{n_i} \frac{v_{R_1}}{4} \right) \quad (2)$$

where there are N paths from s to t, the ith path $R_1, R_2, \ldots, R_{n_i}$ having length n_i, and v_t is the trust in node t by the node carrying out the distributed computation. Although it is a distributed metric (not requiring trusted third parties), the choice of coefficients is arbitrary and does not affect the attack-resistance of the metric. Using a similar argument as for the Beth-Borcherding-Klein metric, this trust metric is also not γ-resistant for $\gamma > 1$.

Jøsang [7]. The metric is based only on the certification paths between s and t, and hence is a 'local' metric. An interesting point is the use of a probabilistic logic which allows one to model uncertainty in recommendations about trustworthiness. Trust values for multiple paths are combined using the *consensus* operator which effectively combines the opinions as if they were observed independently. This allows the metric to be driven arbitrarily close to any desired value, as for the Beth-Borcherding-Klein metric. If, for example, the paths were combined using an operator which required c of them to change significantly before the result changed, then the metric would be c-resistant against attacks with a single forgery.

Maurer [11]. The metric computes a confidence value for a target, using confidence parameters expressed as probabilities on the edges between nodes. Levien [9] shows that the Maurer metric becomes consistent with the shortest path metric when the edge probabilities tend to 0. The Maurer metric, like the Jøsang metric, can be made c-resistant against single-forgery attacks by requiring c independent paths to a trustworthy target.

2.3 More complex trust metrics: 'group' metrics

This subsection presents a class of computational trust metrics with greater attack-resistance, which Levien [8] calls *group* trust metrics. The general idea is to compute the metric over the entire certification graph, to obtain some global solution, rather than a number of bad local approximations (or estimates).

Network Flow (Reiter-Stubblebine [12] and Levien [9]). The Reiter-Stubblebine trust metric uses the concept of network flow in the certification graph, to determine which nodes are deemed trustworthy. Essentially, the 'trust' begins at the source node and 'flows' to all the other nodes. The trustworthiness of the target node is the quantity of trust flow which reaches it. If one assumes that the target nodes have indegree d (corresponding to d certificates being issued per key), then the metric is d-resistant against attacks with a single forgery (a proof is given in Section 2.5).

Levien's Maximum Network Flow trust metric is also d-resistant against single-forgery attacks, and is more resistant to attacks involving the deception of nodes into trusting others (as opposed to stealing their private keys), and is achieved by quickly decreasing the capacities of nodes with increasing distance from the source node.

2.4 A Lower Bound

Using our terminology, we can more precisely restate Levien's main lower bound result on trust metrics for certification graphs. The following lemma provides the basis for the lower bound, essentially stating that if the attack graph is isomorphic to the original graph modulo unreachable nodes (since the names are not important), then no trust metric can notice the attack.

Lemma 1 *If M is a trust metric and $G' \simeq G$ (modulo unreachable nodes), then $M(G) = M(G')$.*

We are now in a position to restare the main theorem of [9].

Theorem 1 *Let G be d-connected. Then no trust metric is γ-resistant against 1-attacks on G, for $\gamma > d$.*

Proof. Consider the 1-attack G', where all predecessors of some target v are victims. For each victim u, replace the edge (u, v) with the edge (u, x) to the forgery x. The attack costs at most d, and $G' \simeq G$. □

One should note that when we say that γ-resistance is a lower bound, this means that no metric can resist successful attacks of cost $< \gamma$. Hence an upper bound

result on the attack-resistance provides a lower bound on the cost of a successful attack.

2.4.1 Attacks with Multiple Forgeries

Previous work has only considered the class of 1-attacks, *i.e.* those which introduce a single forgery. A natural progression is to consider the more general class of α-attacks for $\alpha \geq 1$. We begin with a theorem which bounds both from above and below, the attack-resistance of a trust metric on such attacks, given its resistance against 1-attacks:

Theorem 2 *If a trust metric M is γ_1-resistant to 1-attacks, then M is γ_α-resistant to α-attacks, where $\gamma_1/\alpha \leq \gamma_\alpha \leq \gamma_1$. Alternatively, $1/\alpha \leq \gamma_\alpha/\gamma_1 \leq 1$.*

Proof. We need only consider the class of 2-attacks, since the proof easily generalises to other attacks. Considering the two inequalities separately:

1. In the worst-case, the attacker must add γ_1 victims to create another 1-attack (since the 'union' of two 1-attacks is a 2-attack), so the 2-attack requires $2\gamma_1$ victims and hence costs γ_1 (*i.e.* M's attack resistance does not increase, and $G' \simeq G$). This case is shown in Figure 2a).

2. In the best-case, the attacker need not add any new victims, for a cost of $\gamma_1/2$ (an upper bound for M on 2-attacks, though $G' \not\simeq G$ so it may be possible to detect). This case is shown in Figure 2b).

□

Now consider the negative result, that a metric is *not* γ-resistant for some class of attacks. Then Theorem 2 tells us that if a trust metric M is not γ-resistant to α-attacks, then it is not γ-resistant to $(\alpha + \varepsilon)$-attacks for $\varepsilon \geq 0$. In the case where $\alpha = 1$, we simply say that M is not γ-resistant.

Finally, combining this with Levien's original result, we obtain a simple lower bound for the general class of attacks:

Theorem 3 *Let G be d-connected. Then no trust metric is γ-resistant against attacks on G, where $\gamma > d$.*

Proof. The proof follows from Theorems 1 and 2. □

As we shall see in the next section, this lower bound is tight in the class of 1-attacks, but not for the general class of α-attacks. Hence a natural open question arises: is there a metric which is *uniformly-resistant*, *i.e.* for some fixed γ, is the metric γ-resistant to α-attacks for all $\alpha \geq 1$?

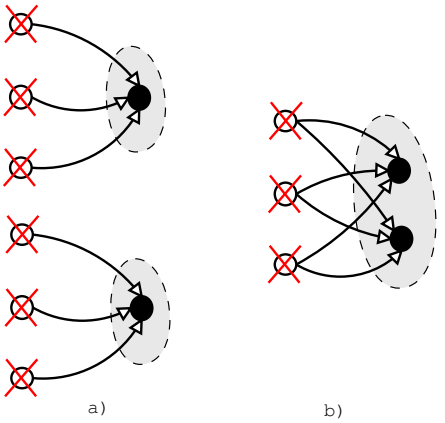

Figure 2. The a) worst-case and b) best-case 2-attacks on a graph. The crosses represent victims and the nodes on the right partition are the forgeries

The search for tighter bounds. We would like to know if it possible to tighten the lower bounds, for the case of α-attacks. Firstly, note that Theorem 1 is a special case of the following lemma:

Lemma 2 *Let $f_k(G)$ be the number of victims required to insert k forgeries into G, to obtain $G' \simeq G$. Then there exists a k-attack G' which costs $c \leq f_k(G)/k \leq d$. Hence no trust metric is γ-resistant against k-attacks on G for $\gamma > c$.*

If G is not d-connected, then $f_k(G) < d \cdot k$, which provides an even tighter lower bound in the case of a particular graph G. Hence an open question is how does the structure of G affect the current lower bound of $\gamma > d$ for α-attacks?

As an example, consider a set V of victims. We say that V *controls* k forgeries if the removal of V disconnects k nodes from G. Then $f_k(G)$ is the smallest number of victims which controls k forgeries. Figure 3 illustrates this with two simple graphs where the same victim set controls different numbers of forgeries. We find that $f_1(G)$ equals the minimum indegree of a target node, so if G is d-connected then $f_1(G) = d$.

2.5 Upper Bounds

In this subsection, we consider the maximum network flow trust metric of Levien-Aiken [9] and outline an upper bound on its attack-resistance. To recap, each node u has a *capacity* $C(u)$, and a target is accepted by a source node if the capacity is above a fixed threshold level. Clearly, the attack-resistance of the metric depends on the distribution of capacities within the

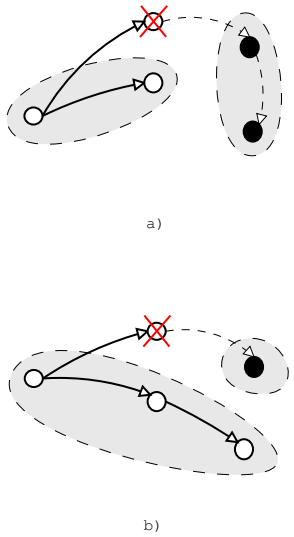

Figure 3. Illustrating Lemma 2. The set V of victims (with crosses) controls a) 2 forgeries, and b) 1 forgery (since control of the other node is not 'inherited' by V). It also illustrates that nodes disconnected from G can be considered to act as victims

graph. Increasing the capacities toward the source intuitively requires more costly attacks, since attacks on randomly-chosen victims are unlikely to be successful. Hence the metric is constrained only by the indegree of the target node. We outline the metric's resistance to 1-attacks below:

Theorem 4 *Let G be d-connected. Then the maximum network flow trust metric over G is optimal for the class of* 1*-attacks.*

Proof. (Outline from [9]) Let the set of victims be V. Define a source node s to be *susceptible* if there exists a node $v \in V$ with capacity $C(v) > 1/d$ (*i.e.* attacking v allows control of at least $1/d$ of the flow from s). The number of susceptible nodes can be bounded from above, hence one can bound the total network flow from victims by considering the capacity of nodes in V, since V is a cut of G'. By considering the fraction of source nodes which can accept the forgery, we find that a successful 1-attack costs at least d, so the metric is d-resistant against 1-attacks on G. □

The trust metric is optimal for 1-attacks, but unfortunately no such result is known for the general case of α-attacks.

2.6 Open Questions

This subsection briefly presents some more general questions arising from our analysis of trust metrics.

We would like to know if there is a relationship between the attack-resistance of consistent metrics, for example 'if M_1 is consistent with M_2 and M_1 is γ-resistant, then M_2 is γ-resistant. Also, if M_1 is consistent with M_2 and M_2 is not γ-resistant, then M_1 is not γ-resistant.'

Resistance scalability. Rather than considering a metric's resistance to a particular class of attack, it may be of use to consider its resistance as the number of forgeries increases. The alternative interpretation of the lower bound of Theorem 2 suggests a notion of the *resistance scalability* of a metric, and hence a related open question is on the existence of a metric which uniformly scales well, *i.e.* has $\gamma_\alpha/\gamma_1 \in \Theta(1)$ (where the metric is γ_α-resistant against α-attacks). Theorem 2 says that no trust metric can do better than 1-resistance scalability, and that there exists a trust metric with $(1/\alpha)$-resistance scalability.

Resource-bound trust metrics. In the certification graph formulation, assuming the certification graph is d-connected corresponds to d certificates being issued per key. It is interesting to consider the equivalent constraint in the recommendation graph formulation. Rather than considering bounded indegree graphs, it may make more sense to consider graphs with a bounded number of edges, corresponding to entities which can store a limited number of recommendations. An interesting open question is what effect do different resource-bounding models (such as a fixed number of edges) have on the fundamental at tack-resistance of a trust metric? (*e.g.* 'How resistant can a metric be on a graph with d edges?')

3 Applications of Trust Models

In this section we discuss some future applications which we feel will benefit from a decentralized trust model. The importance being placed on the development of these applications provides a major incentive to understand and develop secure, attack-resistant computational trust metrics.

- *Routing and Naming Services.* Peer-to-peer routing overlays based on distributed hash tables such as Kademlia, Pastry and Chord are designed to operate in global-scale, open distributed environments, making them particularly good candidates

for the application of trust models. Ad hoc routing is another important application, since the nodes themselves become part of the network fabric and act as routers, and one cannot assume the existence of a trusted third party. The Border Gateway Protocol (BGP) is the standard inter-AS routing protocol deployed on the Internet (the best example of an open decentralized system) yet, surprisingly, it appears to have no notion of trust in that routers blindly accept routing tables from other routers. The goal of a computational trust model in routing is conceptually simple: to minimize the packet loss for nodes wishing to send data. Unfortunately, this is not as easy as one might hope. Even simple trust models for wireless ad hoc network routing such as in [6] can actually make the system less predictable and thus more open to attack.

In much the same way that BGP has little notion of trust, DNS and other naming services are too open to attack. Levien [10] presents a distributed naming service where each server provides a (possibly false) mapping from names to public keys, and the goal is to select servers from which to take a majority vote on the mapping, *i.e.* the trustworthy servers. The attack-resistance properties of the network flow trust metric allow the number of bad servers selected to be bounded, independent of the number of bad servers in the network.

- *Mobile Code.* The downloading and execution of mobile and shared code presents an obvious problem - how can I trust that this code will do what I think it will? This problem is interesting, in that it is amenable to both 'hard' and 'soft' security approaches. Proof-carrying code [5] allows one to construct proofs as valid typed λ-calculus statements, although many properties which one would like to be able to confirm are impossible to 'prove' in this way. A computational trust model would allow one to reason about the source of the code with the aim of identifying those participants who may provide untrustworthy code.

  An interesting use of trust arises at the intersection of mobile code and routing in *active networks*, where the routers essentially follow network code contained in the headers of packets. The trust issue is essentially in deciding whether or not to execute the code, and how this affects the routing properties of the network.

In summary, the main uses for computational trust models appear to be in providing incentives for collaboration and participation, and identification of (to avoid and exclude) misbehaving (or untrustworthy) agents.

4 Conclusion and Future Work

We have presented a precise notion of attack-resistance for trust metrics, developed from the work in [9], and presented simple bounds using a restricted notion of attack-resistance. This notion needs more work; for example to consider the case where not all nodes are equally-easily attacked and to consider the case where nodes are picked at random (as opposed to the worst-case analysis presented here).

This is an interesting area for future work. In addition to the more specific open questions in this paper, we would like to develop a more detailed understanding and classification of the security properties of trust metrics, and, in the distributed case, bounds on their computational and network complexity.

References

[1] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *New Security Paradigms Workshop*, 1997.

[2] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *CIKM*, pages 310–317, 2001.

[3] T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. In *Proc. 3rd European Symposium on Research in Computer Security – ESORICS '94*, pages 3–18, 1994.

[4] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proc. 1996 IEEE Symposium on Security and Privacy*, 1996.

[5] J. Feigenbaum and P. Lee. Trust management and proof-carrying code in secure mobile-code applications. In *DARPA Workshop on Foundations for Secure Mobile Code*, 1997.

[6] D. Johnson, D. Maltz, and J. Broch. DSR: A dynamic source routing protocol for multihop wireless ad hoc networks, 2001.

[7] A. Jøsang. An algebra for assessing trust in certification chains, 1999.

[8] R. Levien. Attack-resistant trust metrics. *PhD Thesis (in preparation)*. `www.levien.com/thesis/`, 2002.

[9] R. Levien and A. Aiken. Attack-resistant trust metrics for public key certification. In *7th USENIX Security Symposium*, pages 229–242, 1998.

[10] R. Levien and A. Aiken. An attack-resistant, scalable name service. `www.levien.com/fc.ps`, 2000.

[11] U. Maurer. Modelling a public-key infrastructure. In *European Symposium on Research in Computer Security*, 1996.

[12] M. Reiter and S. Stubblebine. Toward acceptable metrics of authentication. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 10–20, 1997.