Tolerating Penetrations and Insider Attacks by Requiring Independent Corroboration

Clifford Kahn
EMC Corporation
Hopkinton MA 01748
+1 800 445 2588 ext. 2010
kahn_clifford@emc.com

ABSTRACT

We describe an approach to building a system that meets its requirements even when some central components are successfully penetrated and/or when insiders attack the system. This goal is a key but elusive facet of information survivability.

Our approach relies on independent corroboration, a form of redundancy. Corroboration is easy to pin down; independence is not. How can software judge whether two principals are independent? This paper begins to address the problem. We analyze the word "independence" and find that independence is not absolute, but relative to one's interests; that independence judgments are closely tied to trust; that independence judgments are based largely on known connections between the principals. We then take a two-pronged approach. The first prong is a formal, Bayesian probabilistic model of a system that uses independent corroboration to tolerate compromise. The second is a pragmatic investigation of how independence information may be imported from existing authentication data, and a preliminary look at how knowledge of independence may be dynamically obtained from third parties.

Keywords

Trust models, fault tolerance, security, information survivability.

1. INTRODUCTION

One facet of information survivability may be called *compromise tolerance:* the ability of a system to work acceptably even when components have been compromised. This is akin to fault tolerance: roughly, the ability of a system to meet its requirements even when components suffer faults.

But compromise is different from ordinary faults. First, a compromised component may be malicious. It may be running the adversary's software. Second, fault tolerance generally assumes that faults are independent. Compromises are not independent. A set of redundant servers may be fault-tolerant, but if an adversary gains access to an administrative account on one server, he or she can probably bring down the whole set. Different engineering techniques and a different way of thinking are required to create a compromise-tolerant system.

Of course, not only components, but also human operators may be compromised. We can consider them to be part of the system too, using a larger sense of "system." One wants the system to function properly even when members are compromised. (We shall use the passive form "is compromised" whether a person is compromised by a third party, as through bribe or blackmail, or becomes malicious on his or her own, as from disgruntlement or temptation.)

In our model, the *system* consists of *agents* that operate in some world and that express beliefs to each other about their world. We think of an agent as a person or a running program. Beliefs may include instructions ("I believe that you should do R"). The agents may also denounce each other ("I have observed b acting suspiciously and believe that b is compromised").

Before acting on one of these received statements, an agent must decide whether to believe it. What if the author of the statement was compromised? *Independent corroboration* is the solution. If b says Q and c also says Q, and c is independent of b, then Q is far more credible, far safer to believe.

But how do we determine whether two agents are independent? This is hard, and a part of this paper addresses the question on a very practical level. In sum, we claim that human judgments of the independence of two principals a and b are based on five factors: (1) the interests relative to which the judgment is made, (2) the level of trust placed in a, (3) the level of trust placed in b, (4) any known connection between a and b that might compromise their independence, and (5) barriers that make it less likely for a and b to collude.

A stylistic note: This paper uses the first person singular ("I") to refer to a hypothetical, reasonable person. This paper uses the first person plural ("we") to refer to the author(s).

The following sections consider some related work, the words "trust" and "independence," and how one goes about gauging independence. Then we develop a formal model of independent

This paper describes results of the Methodology for Trust for Network Survivability project. The work was done at The Open Group Research Institute, Cambridge, Massachusetts.

Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-97-1-0248. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation hereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies and endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Rome Laboratory, or the U.S. Government.

1998 NSPW 9/98 Charlottsville, VA, USA
© 1999 ACM 1-58113-168-2/99/0007...$5.00

corroboration, presented with a running example. We reflect on the model, on whether it is plausible and on enhancements it needs. We consider how to derive independence information from existing data sources, and then consider how to obtain independence information from third parties. We sketch an algorithm for computing our trust model's result with better efficiency.

2. RELATED WORK

We are not aware of other work on formalizing independent corroboration. But there are a number of other approaches to tolerating compromise of information sources. All of the following approaches, and our own work, assume isolated compromise—that other measures make it impossible for an adversary to suddenly get control of everything, and that there are some limits to the extent of a plausible conspiracy.

Voting. If one requires an absolute majority vote of some set of agents, and counts non-voters as nays, then the vote is compromise tolerant. Compromising or squelching a minority of the agents will not change the vote.

General interactive consistency. If a number of agents must reach agreement, then Byzantine agreement and related algorithms let them do so without appointing a single trusted vote counter. These algorithms can be used, with care, to build compromise-tolerant systems. Two examples follow.

Byzantine quorum systems. Quorum systems are a technique for implementing fault-tolerant replicated data. Quorum systems have been extended to environments in which some data repositories can be arbitrarily corrupted.[9] Still, as a practical matter, if the repositories have identical implementations and the same administrators, then they share many of the same vulnerabilities.

State machine replication. A service is implemented with multiple identical, deterministic servers, each initialized to the same state. A Byzantine protocol ensures that all servers receive the same inputs from clients. Clients conduct a vote among the servers to decide which servers to believe.

Use of general interactive consistency or Byzantine algorithms does not guarantee compromise tolerance. One must also use secure communication, for example.[6]

Web of trust. If principal A authenticates principal B through a chain of intermediaries, each vouching for the key of the next, then each intermediary is a point of vulnerability. If compromised, any intermediary can pass off an impostor as B. A solution to this is to have redundancy. One can require two non-overlapping chains. More generally, one can require a *trust mesh* rather than a chain, and have criteria for acceptance of a mesh.[11]

Don't trust, verify. In some applications agent A makes an assertion to agent B, who verifies it before acting on it. If A lies it cannot cause much damage. At worst it causes B to waste resources checking on a false assertion.

One application to which this approach applies is a system for tracing attacks through the Internet (Dynamic, Cooperating Boundary Controllers, developed by Boeing and sponsored by DARPA). If a detector observes an attack, it notifies its topologically neighboring detectors. Each of them studies the packets going by, looking for signs of the same attack and taking action if it sees the signs. If a detector makes a false report, then its neighbors will find nothing to confirm it. So they will, in effect, squelch the report. Little harm will be done. Thus such a tracing system can tolerate compromised detectors.

Separation of duty. An old and widely practiced idea is separation of duty among humans. This limits the damage a compromised human can do. For example, an employee cannot approve his or her own expense report.[16]

3. DEFINING "TRUST"

"Independence" and "trust" are appealing, commonsense words that unfortunately cover too much territory.

If I trust my car, I am confident it will not break down and strand me. If I trust my poker partners, I am confident they will not cheat. If I trust my auto mechanic, I am confident both that he or she will not cheat me and that he or she will do good work.

Abdul-Rahman and Halles[1] use "trust" to denote many kinds of trust. But for analyzing independence, we find it helpful to narrow things down and worry only about *betrayal.* We define *betrayal* as when someone or something a user trusts acts as an adversary to the user. (The ordinary sense of betrayal usually involves a broken promise; we do not require that.)

Our concern in this paper is only with betrayal, and not with other failures. We want independence because two independent principals are less likely to both betray a user.

I can have other kinds of trust in humans. I can trust a movie reviewer to recommend movies I will like. But independence of movie reviewers is different. To say that two movie reviewers are independent is to say that they form their own opinions and sometimes disagree. Even if independent, they may both recommend a movie that I dislike. One could study how to combine the recommendations of experts to produce an aggregate recommendation. But this is outside the realm of computer security.

I can have other kinds of trust in software components also. I may trust a Web site to present current and accurate stock-market quotes. To say that two software components are independent in terms of (for example) their timeliness and accuracy would mean that the components were unlikely both to make the same error at the same time. But evaluating and taking advantage of that kind of independence and trustworthiness is the domain of Software Reliability.[8] Reliability theory makes different basic assumptions from computer security about what combinations of failures are likely to occur. Anything that can be handled as a reliability problem instead of a computer security problem should be.

So in the balance of this paper "trust" will mean "trust not to betray." This narrow reading of "trust" may be better for computer-security work in general.

4. RELATIVITY OF INDEPENDENCE

Relevant dictionary definitions of "independent" are:

> *"not subject to control by others"*
>
> *"not looking to others for one's opinions or for guidance in conduct"[17]*

By extension, when we say that two people are independent of each other, we mean that they are not subject to each other's control, they do not look to each other for their opinions or guidance, and there is no third party from whom both take opinions or guidance or to whose control both are subject.

Still, everyone has influences, and any two people have influences in common. What matters is whether the influences are *relevant* and whether they are *too strong*.

Relevance depends on one's interests. If I work for a large company, then two departments of the company may be fairly independent with respect to my department's interests. But they will not be independent with respect to a competitor's interests. *Independence is relative to a set of interests.*

Another relevant dictionary definition of "independent" is the probabilistic definition. Applying this to trust and betrayal, we propose:

Two agents a and b are independent relative to an agent c if the circumstances under which a would betray c have no correlation with the circumstances under which b would betray c.

This definition is consistent with the foregoing definitions, and somewhat more precise. It is not mathematically precise, because it uses "correlation" informally. To use "correlation" precisely, we would have to posit a probability distribution over circumstances. Some Bayesians might be willing to posit that distribution. But it is over an infinite domain, and it is not clear what sort of finitely representable probability distribution might serve, so we do not posit one. Thus we do not have mathematical precision, but we find the definition useful.

This exacting kind of independence is impossible: there is always a nonzero chance that two agents will collude. But one can come close.

5. GAUGING INDEPENDENCE

If software is to gauge the independence of keys or principals, administrators will have to supply information about this. The notation in which administrators express their knowledge of independence must make sense, must be intelligible to those who use it. It is not enough to spray-paint on a nice graphical user interface (though we have seen that fallacy). Rather, suitability to the administrator's task is a fundamental property.[18]

5.1 Levels of Trust

The more I trust two agents, the more I believe they are independent (other things being equal). If I trust a and b highly, and c and d less, then as a rule I will think c and d are more likely to collude against me than a and b are. This is common sense, and I do not need an independence metric to tell me this.

So we posit a real number representing the trust level of each agent, something akin to a security clearance. But that is not enough.

5.2 Connections

I also judge independence by whether or not there are known connections that could compromise the independence of two agents. If two people are married or in the same immediate family I do not assume they are independent for most purposes, which is why companies forbid one to work for his/her spouse. If they work for the same company then they are not independent from the viewpoint of a standards body. And if two keys are held by the same person, then of course the keys are not independent.

In our model, each agent has a set of ***influences*** that may lead it to compromise. An influence may be:

- an organization, such as the person's employer or the machine's owner
- a marriage, friendship, or similar association
- a vulnerability, whether known or hypothesized, such as a particular program

Of course, different agents may be subject to the same influence. But an influence need not influence every agent the same amount. For example, if you a citizen of Country X, then I will assume Country X influences you. If your parents are from Country X and you are not, I may suspect you still have more loyalty to Country X than most people do, but not as much as if you were a citizen.

Therefore the effect of each influence on each agent is weighted: for each agent and each influence there is a real number in $[0,1]$ denoting the weight of the influence over the agent. If a and b both have strong affiliations with an organization J, then a and b will not (as a rule) be considered independent.

For simplicity, in our model agents do not directly influence other agents. Any potential collusion among a set of agents must be expressed by creating a pseudo influence that affects all of them.

Likewise, in our model organizations do not influence other organizations. So if in the world organization J influences organization K and K in turn influences agent a, then in the model J's influence on a should be direct. This flattening makes the model much simpler to analyze. However, this flattening is possible only if the influences are partially ordered.

We do not model how the analyzer comes to know what influences another agent. There is no notion of learning about influences from a third party who itself might or might not be telling the truth. The analyzer has beliefs about the influences affecting all of the other agents, but we do not have a model of sharing them. This, of course, is simplistic and begs for refinement.

5.3 Barriers

I also judge independence by whether I know of barriers that would tend to make it harder for two parties to collude. If I know that a particular compromising connection does not exist, that is one kind of barrier. Other barriers are created by explicit policy.

Common barriers of the first kind include:

- Being different people (since one person can hold many keys and therefore be many principals)
- Having different employers

A well-known barrier of the second kind is the Chinese Wall.[2]

Depending on the need, I may investigate people for particular connections that could compromise their independence. Judges, for example, ask jurors whether they know the defendant or any of the lawyers. In a corporate setting I might ask people whether they are family members, business partners, etc. I might check public records. For highly trusted positions, this much digging may be necessary to avoid conflicts of interest, fraud, and such. If

I have investigated and found no compromising connections, then I am more confident no such connections exist.

A barrier, then, may be merely the confirmed absence of a particular kind of connection. But as well, organizations actively erect barriers such as Chinese Walls to make compromising connections less likely. Special prosecutors engender some trust because they are relatively protected from the administration. Consumer Reports magazine refuses advertising.

5.4 Set of Interests

As Section 4 argued, independence is relative to a set of interests. We model the set of interests as the levels of trust the analyzer has in the different influences. Suppose c is gauging the independence of a and b. Suppose a and b are both affiliated with J. For example, J may be the country of which a and b are citizens.

If c does not trust J highly, then c's independence gauge goes down.

If c trusts J highly, then c's independence gauge stays the same.

If c trusts J moderately, then c's independence gauge goes down, but not as much as in case #1.

5.5 Summary

My gauge of two people's independence is a function of how trustworthy I think each of them is, whether I know of or suspect any compromising connections, what barriers I know of between them, and the set of interests relative to which I am judging.

6. RUNNING EXAMPLE

We are applying this approach to intrusion detection, to insure that an intrusion detection system works correctly even if components are compromised. The approach applies especially to the problem of detecting and analyzing a widespread intrusion, one that spans organizations.

Intrusion detection components are subject to compromise, like anything else. False reports from an intrusion detector can be extremely damaging, especially if they trigger automatic responses.

Moreover, when an attack is widespread, some of the detectors may belong to organizations that one does not completely trust. Nevertheless one needs their reports. One needs corroboration. The question is how to combine the inputs from various detectors in order to decide what is really going on, even in a case where some intrusion-detection components are compromised.

For the example, my organization gets information from four other organizations. The first two, i and j, are close business associates with my organization, so they are relatively trusted. The second two, k and l, are reputable but they lack close ties to my own organization and so are less trusted.

Some of these organizations run several intrusion detectors in different locations. It is generally reasonable to assume a degree of independence among detectors in different locations. One of the detectors belongs to a joint project between j and k, so might be influenced by either organization.

Organization	Detector
i	a
j	b
j,k	c
k	d
l	e

Table 1: Example Organizations and Detectors

7. TRUST NOTATION

The model is of a system consisting of agents. Think of these agents as people, machines, and processes. They correspond in important ways to principals. But they are more than principals, in that we are modeling their beliefs.

Some or all of the agents have beliefs about the other agents' trustworthiness and affiliations. And each agent's beliefs in this regard reflect the agent's own interests. Thus, the model reflects the principles of Section 5.

The agents also have beliefs about the world, and they communicate these beliefs to each other. The point of the model is to allow an agent to decide whether an assertion made by one or more other agents is true, or whether those agents are lying. For simplicity, the model does not permit mistakes. All falsehoods are lies.

The model assumes that all communication is authenticated, so there is no doubt about which agent made a particular assertion. When thinking about implementations, you may wish to assume a one-to-one mapping from agents to public keys.

7.1 Trust and Independence Metrics

A system consists of a nonempty set A of agents and a set I, disjoint from A, of influences. To avoid self-reference and to generally simplify the math, we stipulate that the agent doing the analysis is *not* a member of A. We refer to the agent doing the analysis as the *analyzer*.

The analyzer holds a set of beliefs about the trustworthiness and independence of the other agents. This set of beliefs consists of:

- A pair of functions $t_A{:}A \rightarrow [0,1]$ and $t_I{:}I \rightarrow [0,1]$ that map each agent or influence into a number between zero and one; we call this number the "trust level". 1 is the highest trust.

  If the analyzer were strongly affiliated with an organization $o \in I$, then I would expect the analyzer's trust in o, $t_I(o)$, to be very large. This reflects the principle that independence is relative to a set of interests (see Section 4, Relativity of Independence).

- A function $w{:}I \times A \rightarrow [0,1]$ that maps an agent and an influence into a number between zero and one; we call this number the "weight". 1 is the strongest influence.

- A function $s{:}A \times A \rightarrow [0,1]$ that maps a pair of agents into a number between zero and one that measures how much of a barrier is believed to exist between those two agents. ("s" stands for "separator".) The bigger the number, the stronger the barrier. One can express this function as a matrix; we envision it as a sparse matrix, with most of the entries containing a default, small value. Again, a barrier is either the fact that a particular connection does *not* exist (for example, that x and y do not work together), or that some explicit barrier has been created. This function is symmetric: $s(x,y) = s(y,x)$.

$I=\{i,j,k,l\}$ $A=\{a,b,c,d,e\}$	$t_A(a)=$ .999	$w(i,a)=$ .5
	$t_A(b)=$ .999	$w(j,b)=$ .5
All $s(x,y)$ *values .99999*	$t_A(c)=$ .99	$w(j,c)=$ .5
	$t_A(d)=$ .99	$w(k,c)=$ .5
	$t_A(e)=$ .99	$w(k,d)=$ .5
	$t_I(i)=$ .999	$w(l,e)=$ .5
	$t_I(j)=$ .999	
	$t_I(k)=$ .99	*All other w values zero*
	$t_I(l)=$ .99	

Example 1: Trust and Independence Beliefs

See Example 1.

8. SEMANTIC MODEL

We endow this notation with formal semantics by mapping this set of beliefs into a Bayesian network[10].

8.1 Bayesian Networks

A Bayesian network is a directed acyclic graph (DAG) whose nodes are random variables. In our model all nodes are Boolean.

The root nodes represent random variables that are independent of each other, and for each of these nodes the network specifies a probability. These probabilities are known as the *priors*. For each nonroot node, the network specifies the conditional probability given each possible combination of its direct predecessors.

Certain independence assumptions hold, but we will not state all of them here.

Bayesian networks allow one to calculate the conditional probabilities of the nodes in the network given that the values of some of the nodes have been observed.[3]

8.2 Inputs

Consider a nonempty set of agents A, a set of influences I, and functions t_A, t_I, w, and s as described above.

Consider an assertion Q whose truth the analyzer wishes to evaluate. We assume that adversaries would be motivated to assert Q if Q were false. (Otherwise there is no need for the model.) Let us assume that in the analyzer's judgment, the prior probability of Q is $P(Q)=\varepsilon>0$. The model does not say how to come up with this probability nor whether or not the probability varies from one assertion to another.

Let us also assume that in the analyzer's judgment, the prior probability, given that Q is true, that an uncompromised agent will know Q is true and will announce this to the analyzer is δ. Formally, $P(S(x,Q)|Q,\neg F(x))=\delta$; we will explain this notation below.

8.3 Simplifying the Inputs

In our model there are three ways an agent can come to be compromised. It can collude with any other agent. It can be influenced by a compromised organization with which it is known to be affiliated. Or it can simply compromise itself, without the involvement of any other agent or organization.

It is convenient to treat all three of these the same way. We fold the collusions and self-compromises into the influences, weights, and trust levels. So we create a new set I', a new function $t':I'\rightarrow[0,1]$, and a new function $w':I'\times A\rightarrow[0,1]$.

Initially, we set $I'\leftarrow I, t'\leftarrow t_I, w'\leftarrow w$.

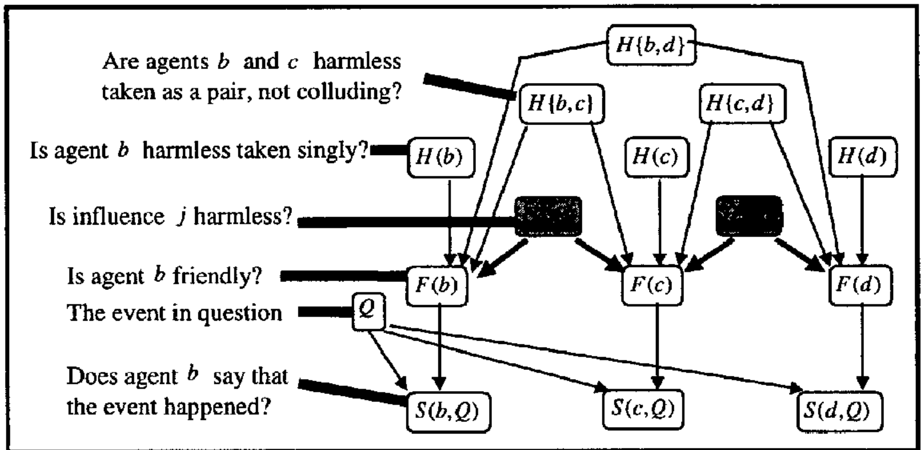
We add each agent $x\in A$ to I'. We set the trust level $t'(x)=t_A(x)$. We set the weight $w'(x,x)=1$. For all other agents we set the weight to zero: $w'(x,z)=0$ for all $z\in A-\{x\}$.

We add each set of two agents to I'. So for each set of two agents $\{x,y\}\subseteq A$, $x\neq y$, we set $\{x,y\}\in I'$. We set the trust level $t'(\{x,y\})=s(x,y)$. If two agents collude, both are compromised, so we set the weights $w'(\{x,y\},x)=w'(\{x,y\},y)=1$. For all other agents we set the weight to zero: $w'(\{x,y\},z)=0$ for all $z\in A-\{x,y\}$.

8.4 Constructing the Network

For each influence $i\in I'$ the network has a unique Boolean node $H(i)$, representing roughly the event that i is harmless, not a compromising influence. This is a root node with prior probability $P(H(i))=t'(i)$. Its intended interpretation follows:

- If i is a pair of agents $\{x,y\}\subseteq A$, then $H(i)$ being false means that x and y *privately collude*. A *private collusion* is when x and y collude on their own, and not because of an influence listed in the original set I. Also, if some influence $j\in I$ affects both x and y, but the analyzer does not know this ($w(j,x)=0$ and/or $w(j,y)=0$), that counts as a private collusion.
- If $i\in I$, then i is one of the original influences known to the analyzer. Then i may represent an organization or vulnerability. $H(i)$ being false means that the organization is


Are agents b and c harmless taken as a pair, not colluding?
Is agent b harmless taken singly?
Is influence j harmless?
Is agent b friendly?
The event in question
Does agent b say that the event happened?
H{b,d}
H{b,c}
H{c,d}
H(b)
H(c)
H(d)
F(b)
F(c)
F(d)
Q
S(b,Q)
S(c,Q)
S(d,Q)

Example 2: Bayesian Network (subset of previous examples)

fairly badly compromised, or that an adversary is attacking the vulnerability.

- Otherwise, i is an agent: $i \in A$. $H(i)$ being false means that i is compromised *spontaneously and singly*, rather than for either of the above two reasons. $H(i)$ being false includes the case that i is compromised because it is subject to an influence without the analyzer's knowledge.

The network has a Boolean node Q; this node represents the event that Q is true. This is a root node with prior probability $P(Q)=\varepsilon$.

For each agent $x \in A$ the network has a unique Boolean node $F(x)$, representing the event that x is friendly, not compromised. Agent nodes are not influence nodes: $\forall x \forall i{:}\, F(x) \neq H(i)$. (But note that each agent $x \in A$ gives rise to two nodes, $F(x)$ and $H(x)$.) For each influence $i \in I'$, the network has an arc $\overline{ix}$, unless $w'(i,x)=0$. The conditional distribution of $F(x)$ given that a certain subset J of its predecessors are compromised is:

$$P\left(F(x)\,\middle|\,\begin{array}{l}J\ \textit{is the set of}\\ \textit{predecessors of } x\\ \textit{that are compromised}\end{array}\right)=\prod_{j\in J}(1-w'(j,x))$$

That is, for all of the influences that are compromised and that affect x, we take the product of x's imperviousness to their respective influences. It is as if x, in order to remain a good guy, had to make it past a number of independent hurdles: each of the bad influences on it. See an example in Table 2.

$H(c)$	$H\{b,c\}$	$H\{c,d\}$	$H(j)$	$H(k)$	P
F	*	*	*	*	0
*	F	*	*	*	0
*	*	F	*	*	0
T	T	T	F	F	$(1-w(j,c))$ $(1-w(k,c))$ $=.25$
T	T	T	F	T	$1-w(j,c)=.5$
T	T	T	T	F	$1-w(k,c)=.5$
T	T	T	T	T	1
* denotes don't care					

Table 2: Conditional Distribution for $F(c)$

For each $x \in A$ the network has a Boolean node $S(x,Q)$, representing the event of agent x saying that Q is true. There is an arc from Q to $S(x,Q)$, because whether Q is true affects whether x says Q is true. There is an arc from $F(x)$ to $S(x,Q)$, because whether x is friendly affects whether x says that Q is true.

The conditional distribution of $S(x,Q)$ is an instance of **Error! Reference source not found.**. The table says that if x is compromised x will certainly say Q. If x is friendly, x will tell the truth about Q or keep silent. It would not be hard to think of a more sophisticated distribution, but we want to start simple.

Q	$F(x)$	P
F	F	1
F	T	0
T	F	1
T	T	δ

Table 3: Conditional Distribution for $S(x,Q)$, where $x \in A$

We hope it is clear that the Bayesian network has the following properties, where x is an agent:

- The probability of x's avoiding compromise given that none of the influences that affect it is compromised and that it is not involved in any private collusion is $t_A(x)$.
- Given that $H(x)$ is true, that $H\{x,y\}$ is true for every agent y, and that exactly one compromising influence, $i \in I$, affects x, the conditional probability of x's being compromised, of $\neg F(x)$, is $w(i,x)$. The stronger the weight of an influence on x, the higher the probability that x will be compromised given that the influence is compromised, other things being equal.
- Whenever an additional influence that affects x is compromised, other things being equal, the conditional probability of x being compromised goes up.

Nevertheless, simplicity is the main virtue of our conditional distribution for an agent staying friendly, $F(x)$. A known problem with this distribution is that we think $t(x)$ should reflect the *unconditional* probability $P(F(x))$. We think this would better reflect the meaning of a human trust judgment.

9. EVALUATING OBSERVATIONS

We present a naïve approach to evaluation, explain its limitations, and then present a more sophisticated approach.

9.1 Naïve Evaluation

Given observed values for zero or more of the S nodes, one computes the posterior probability of Q using standard Bayesian-network techniques.

If this probability is high enough, then one acts as if Q were true. How high a probability is high enough is application dependent.

In our example, $P(Q|S(a,Q)\wedge S(b,Q))$ is much higher than $P(Q|S(a,Q))$ or $P(Q|S(b,Q))$. Thus, as one might expect, I can believe Q with more confidence when a and b corroborate each other.

9.2 Limitations of Naïve Evaluation

The naive approach ignores the likelihood that the adversary has a strategy. They will attack points they believe to be weak, and they will marshal resources.

For example, suppose that I know of no common influences affecting more than one agent. Suppose my trust in the agents is such that if any three of them agree on something, I will believe the three. It is clear that the more agents exist, the more vulnerable I am. If three of seven agents agree on something, that is one thing. If three of a thousand agree, that is a different and much less significant thing. The naive evaluation technique does not capture that difference. I simply ask what is $P\left(Q\middle|S(a,Q)\wedge S(b,Q)\wedge S(c,Q)\right)$, where a , b , and c are the three agents. This question ignores the existence of 997 other agents who have not voiced an opinion. The adversary might have carefully selected three agents that it could compromise.

9.3 Statistical Technique

Standard technique in experimental statistics is to (1) choose a statistic to compute over the observations, (2) choose a desired significance level, which is the probability of seeing a false positive given that that the null hypothesis is true; (3) compute a threshold for the statistic, such that the probability that the statistic will exceed the threshold given the null hypothesis equals the desired significance level. The null hypothesis is the hypothesis that the effect for which one is looking is not there. One minus the statistical significance is called the *confidence*.

For our purposes, the null hypothesis is the hypothesis that Q is false: there is no intrusion or whatever. Next we must define the statistic.

For each nonempty set $X \subseteq A$ of agents, we add a unique node $S'(X,Q)$. For each of the agents $x \in X$ we add an arc from $S(x,Q)$ to $S'(X,Q)$. The conditional distribution of $S'(X,Q)$ is just an AND gate: if all of its immediate predecessors are true, then $S'(X,Q)$ is true with probability 1. Otherwise $S'(X,Q)$ is false with probability 1.

With each of these new nodes we associate a significance value, which is $P\left(S'(X,Q)\middle|\neg Q\right)$. This is computable using standard Bayesian-network techniques.

Our overall statistic, which we call the *spoof score*, is the minimum significance value of the $S'(X,Q)$ nodes that are known by observation to be true. It is not very helpful to think of this result as a probability.

After choosing a threshold, t , for the spoof score, we can create still another Boolean node, r , set up to be true if any $S'(X,Q)$ node whose significance is below the threshold t is true. This is the result node. The analyzer will believe Q if r is true, and not otherwise. So, the overall statistical significance of our evaluation is $P(r|\neg Q)$. Again, this probability can be computed using standard Bayesian-network techniques. We adjust the threshold so that $P(r|\neg Q)$ has the desired value.

9.4 Example

Continuing our running example, and setting the spoof-score threshold to .00001, the following combinations of reports will cause our analyzer to believe Q: a,b; a,c ; a,d ; a,e ; b,d ; b,e; c,e . All supersets of these combinations will also cause our analyzer to believe Q .

So, the analyzer requires two reports from two independent organizations. At least one of the reports must be from the more trusted organizations, i and j . The dual role of agent c , as part of both organization j and organization k , is handled conservatively.

These results are reasonable.

10. WHERE DO THE NUMBERS COME FROM, AND DO THEY MEAN ANYTHING?

Our model has the difficulty of all Bayesian models: the prior probabilities are hard to come by. If these probabilities are too inaccurate, then the outputs of the model are worthless.

For example, what is the prior probability that a particular host will be compromised? It seems impossible to answer that question without more information about what else would be going on at the time.

However, all models are simplifications of reality. A useful scientific question is: Does the model yield empirically accurate predictions?

We offer three views on how it may do that in practice.

10.1 The Model Viewed As an Expert System

Let us suggest an experiment that would test the model.

Devise a qualitative scale of trustworthiness and a qualitative scale of vulnerabilities. One might name the points on the scale, or just use numbers: how trustworthy is organization i on a scale from 1 to 10?

Devise a mapping from these qualitative scales into [0,1].

Take a real set of organizations, principals, and vulnerabilities. Have a knowledgeable informant rate each principal, organization, and vulnerability on the appropriate scale.

Then create scenarios along the following line. What if a,c,d all say that Q happened: should we believe them? If the model's conclusion agrees with the informant's, then the model is succeeding.

Thus, the model's job is to produce the same judgment as a human expert, most of the time, but to produce the judgment faster and without human intervention. The model is, in short, an expert system.

10.2 Plausibility Measures

Much work has gone into the difficulty of estimating prior probabilities. One tack is to move away from numeric probabilities and into other kinds of plausibility measures. Whereas probabilities are drawn from the interval [0,1], plausibility measures can be drawn from any partially ordered set.[4] The set can be discrete, and thus can more directly represent qualitative judgments about the likelihood of an event.

A drawback of discrete plausibility measures is that it may prove harder to find an efficient algorithm for evaluating the network. And because they are more general than probabilities, they are harder to think about. So we thought it best to start with probabilities. Trying out plausibility measures is a logical next step.

10.3 Learning

There is literature on learning algorithms to tune the prior probabilities in Bayesian networks.[7]

Perhaps that approach can be applied here, using cases where ground truth has been established by human investigators. However, the ground data is sparse and so it is not clear whether this approach would succeed or not.

10.4 Sensitivity Analysis of Model

As long as we are working with numeric probabilities, it would be helpful to know how sensitive the results are to the exact values of the prior probabilities. Intuitively, we might say that when it comes to probabilities, .99 is an order of magnitude greater than .9, and .999 is an order of magnitude greater still. We conjecture that in our model, probability estimates need only be in the right order of magnitude. If this can be made precise and proven, then it will be much easier to believe that humans can produce accurate enough probability estimates.

11. ENHANCEMENTS

11.1 Denunciation

It would be useful to allow agents to denounce each other. One agent, especially a human agent, may know that another agent has been compromised. But false denunciations happen all the time, and are an easy way to attack the system.

This seems easy to handle. For any two agents x and y, we create a new node $D(x,y)$, which represents the event that x denounces y. The conditional distribution will be something like Table 4.

$F(x)$	$F(y)$	$D(x,y)$
F	F	.5
F	T	.5
T	F	.5
T	T	0

Table 4: Conditional Distribution of $D(x,y)$

Thus, the event $D(x,y)$ will generally cause me to give less weight to the assertions of either agent, since it casts suspicion on both. But if one agent denounces many others, the probable diagnosis in a typical network will be that the denouncer is compromised. And if many agents denounce one, the probable diagnosis in a typical network will be that the denounced agent is compromised. All of this is as it should be.

11.2 Adding Change over Time to Model

The model does not include time. Trust decisions are made at an instant. But trust levels do change. Some of the factors that make them change must remain outside the model. But other factors are available and could be handled automatically.

For example, an agent whose report proves false should be trusted a little bit less in the future, and one whose report proves true should be trusted a little bit more.

11.3 Dependency of Trust on Purpose

How much I can trust a principal or organization depends on what assertion I am interested in. For example, suppose my company has a joint project p_1 with company c_1 and a joint project p_2 with a company c_2. Suppose c_1 and c_2 are competitors. How much I can trust an assertion by p_1 depends on whether the assertion is about p_1 or p_2.

Our model would be more valuable if it took this dependency into account. As it stands, one must take this dependency into account when fixing the prior probabilities for the model.

11.4 Hedging Probabilities

In case the sensitivity analysis mentioned above find that the model is sensitive to the exact values of probabilities, one could a system for hedging probabilities. In Dempster-Shafer theory, for example, each event is given a range of probabilities rather than a single number.

11.5 Measure of Adversary's Effort

It is hard to tell how much good this model would do in a given case, even if all of the probabilities were accurately estimated. It would be useful to have the model predict how much work the adversary would have to do to defeat the model, to make the analyzer believe something false.

This would have profound affects upstream in the model. For each influence the model would have a cost of compromise. This cost might replace the probability of compromise. Or perhaps the model would have a combination of cost and probability, reflecting the fact that the adversary also lives in an uncertain world and cannot predict his or her costs exactly.

12. EXISTING SIGNS OF INDEPENDENCE

A serious question with the foregoing model or anything in the same vein is: Where would all of the information about independence come from? Practically speaking, we must take advantage of existing authentication infrastructure. Administrators will not add independence information to all the principals by hand. It is too much work.

The following are ways that an administrator might add a lot of independence information in one fell swoop, by taking advantage of existing practice. The information that follows is information about *barriers* among principals.

- An authentication source's administrators may have practices to insure that the source certifies only one key per principal, and that the principals have some measure of independence.

  For example, an employer's certification authority (CA) might certify exactly one key for each employee. Although employees are not necessarily independent of each other, they are independent to a first order: they are different people.

  If this practice exists and the CA software is suitably equipped, the administrator could tell the CA to automatically certify a specified barrier level among all of its principals, except for specific cases where the administrator overrode this level.

- Similarly, an authentication source's administrators may have practices to insure that a particular security group contains only one key for each principal, and that the principals have some degree of independence.

- Administrative practices may insure that the members of group a are independent of the members of group b. This occurs in the world: companies are required to hire auditors, and the premise seems to be that every certified auditor is independent of every client.

  The foregoing are the general cases. Special cases and variations follow.

- In most DCE[15] cells it would probably be somewhat hard to get two keys assigned to the same host. To spoof the cell administrators and acquire multiple keys, what would an adversary say: that his/her host forgot its key? Of course, everything depends on how the cell is run.

- In most Domain Name Service (DNS) domains, it will probably be somewhat hard to get two keys assigned to the same host, assuming the domain uses the forthcoming public-key system DNSsec (RFC 2065).

  This independence is quite weak since two hosts might still depend on the same NFS file server, for example.

- A global certification authority (CA) typically has practices that make it unlikely to sign a false certificate. If so, the key assigned to a principal whose DNS name is `*.W.X` is likely to be independent of the key assigned to `*.Y.Z` if `W.X` ≠ `Y.Z`. This independence inference is stronger if the associated organizations' names are reputable – for example, if they are found on some list of established and non-shady companies, or of government agencies.

So many clear signs of independence can be found in existing security infrastructures, if one selects carefully. The administrator must be able to decide which sources to import such information from, and how completely to trust those sources. Also, the imported information indicates that certain barriers exist to compromising connections. These barriers create some presumption of independence. But the administrator may still know of affiliations that are relevant between particular principals, and must be able to enter this information.

13. SECOND-HAND INDEPENDENCE

Suppose that every principal had sufficiently accurate independence metrics for the principals it knew first-hand – the principals to which it had been manually introduced. This would be enough if one learned everything through direct observation or from these first-hand informants, as illustrated in Figure 1.

In the figures in this section, an arrow represents a belief by the principal at the tail that it can authenticate the principal at the head, and that the principal at the head is fairly trustworthy. (For simplicity this section assumes a yes or no scale of independence and a yes or no scale of trust.) For example, a can authenticate b and c, and trusts them. Arcs represent beliefs about independence. a believes that b and c are independent.

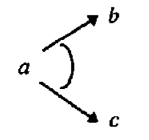


Figure 1: First-Hand Independence

However, what if a meets a principal through an intermediary? Then checking for the proper degree of independence is harder.

In Figure 2, a believes that b and c are independent of each other. But this is not enough. a has no basis, not even a shaky one, for inferring that d and e are independent. If a takes e's assertions to corroborate d's, then a accepts a single point of vulnerability if d and e happen to be the same person, or close associates.

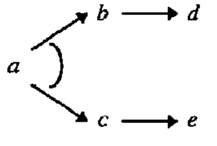


Figure 2: Second-Hand Independence Unknown

In Figure 3, b and c both believe that d and e are independent. a may infer that d and e are indeed independent, and therefore can corroborate each other. There is no single point of vulnerability.

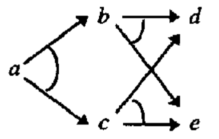


Figure 3: Second-Hand Independence Known

We should note that there is an intellectual tradition around n^th^-hand trust, including Pretty Good Privacy[5] and Reiter and Stubblebine's work on trust meshes.[11][12] We have adopted their goal of avoiding a single point of compromise. However, to the best of our knowledge they lack a real theory of independence, and (again to the best of our knowledge) they have not yet solved the problems just discussed.

We expect that the last example, Figure 3, can be generalized to more elaborate graphs. It also needs extension to handle graded rather than black-and-white trust, and to handle the more sophisticated criteria of independence discussed in the preceding sections. With these enhancements, we would have a solid theory of n^th^-hand independence, which in turn would allow principals to share information about other principals' trustworthiness and independence. This is a critical step toward practicability for compromise-tolerant systems.

14. EFFICIENT COMPUTATION

A straightforward implementation of the computations mentioned above would be inefficient in practice.

We do not propose to represent all of the S' nodes and the r node of the Bayesian network as objects in memory. Rather, when a set X of agents asserts Q, we will construct the $S'(X,Q)$ node on the fly. We will estimate that node's significance value. This significance value equals the spoof score, as we will show. We will compare the estimated spoof score to the estimated value to the threshold, and thus determine whether an arc from that node to r probably exists.

There are three main steps: determining the significance value of the S' node, determining the spoof score, and determining the threshold. We treat them in turn.

14.1 Determining the Significance Value of an S' Node

Given a set of agents $X \subsetneq A$, the significance value of the $S'(X,Q)$ node is, by definition, $P\left(S'(X,Q) \middle| \neg Q\right)$.

Lemma 1 $P\left(S'(X,Q) \middle| \neg Q\right)$ =

$$\sum_{J \subseteq I'} \left(\prod_{j \in J} (1 - \iota'(j)) \prod_{j \in I'-J} \iota'(j) \prod_{x \in X} \left(1 - \prod_{j \in J} (1 - w'(j,x)) \right) \right)$$

Proof: If $J \subseteq I'$ is a set of influences, let $\lambda(J)$ mean that J are the compromising influences. So $\lambda(J)$ is equivalent to $J = \{ i \in I' | \neg H(i) \}$.

$$P\left(S'(X,Q) \middle| \neg Q\right)$$

= by the Law of Total Probability

$$\sum_{J \subseteq I'} P(\lambda(J))\, P\left(S'(X,Q) \middle| \neg Q, \lambda(J)\right)$$

= because, by the structure of the network, all of the S nodes are conditionally independent of each other given the values of the H nodes

$$\sum_{J\subseteq I'} P\big(\lambda(J)\big)\prod_{x\in X} P\big(S(x,Q)\big|\neg Q,\lambda(J)\big)$$

= by the conditional distribution of $S(x,Q)$

$$\sum_{J\subseteq I'} P(\lambda(J))\prod_{x\in X} P\big(\neg F(x)\big|\lambda(J)\big)$$

= by the conditional distribution of $F(x)$

$$\sum_{J\subseteq I'} P(\lambda(J))\prod_{x\in X}\left(1-\prod_{j\in J}\big(1-w'(j,x)\big)\right)$$

= by the definition of $\lambda(J)$ and by the independence of the H nodes

$$\sum_{J\subseteq I}\left(\begin{array}{l}\prod_{j\in J} P(\neg H(j)) \prod_{j\in I'-J} P(H(j)) \\ \prod_{x\in X}\left(1-\prod_{j\in J}(1-w'(j,x))\right)\end{array}\right)$$

= by the distribution of $H(j)$

$$\sum_{J\subseteq I}\left(\prod_{j\in J}(1-t'(j)) \prod_{j\in I'-J} t'(j)\prod_{x\in X}\left(1-\prod_{j\in J}\big(1-w'(j,x)\big)\right)\right)$$

We can estimate this sum by sampling a random set of terms, taking the mean of the sample as an estimator of the mean of the terms of the sum, and multiplying this estimator by the number of terms, which is $2^{|I'|}$. To know how many terms we need in the sample, we choose a desired confidence, such as 99%, decide on the width of the desired confidence interval, and by standard population-sampling techniques[13] compute the number of terms needed.

14.2 Determining the Spoof Score

Lemma 2 If $X \subseteq Y \subseteq A$, then $P(S'(X,Q)|\neg Q) \geq P(S'(Y,Q)|\neg Q)$. More agents asserting Q can only lower the significance value, increasing confidence.

Proof: If we look at Lemma 1 we see that the only effect of adding another agent y to the set X is to add another factor of the form $1-\prod_{j\in J}(1-w'(j,y))$ to each term of the significance value. Since the w' values are all between zero and one, this new factor must be between zero and one. Since the other factors are also nonnegative, adding a factor between zero and one leaves the value of the term the same or smaller. Therefore the value of the sum is the same or smaller.

We defined the *spoof score* as the minimum significance value of the $S'(X,Q)$ nodes that are known by observation to be true. Suppose the set of agents that say Q is Y: $Y=\{y|S(y,Q)\}$. Then, for any subset X of Y, $S'(X,Q)$ must be true, by the structure of the Bayesian network. For any set of agents Z that is not a subset of Y, $S'(Z,Q)$ must be false, by the structure of the Bayesian network.

Combining this with Lemma 2, we see that the S' node that is true and has the lowest significance value is $S'(Y,Q)$. And its significance value is the spoof score, by definition.

Therefore we need not do any additional computing to determine the spoof score.

14.3 Determining the Threshold

As Section 7.3 discussed, we need to set a threshold for the spoof score. Again we use a sampling technique.

For each sample, we randomly set each of the root nodes in the Bayesian network, according to each node's prior probability, except that we set Q false. Now we identify the set of agents that assert Q: $Z=\{z\in A|S(z,Q)\ \textit{is true}\}$. $S'(Z,Q)$ is the S' node whose significance score is lowest, as we showed in Section 14.2.

Sample repeatedly, and then set the threshold such that the desired fraction of the sample is below the threshold, according to the significance level chosen. This is a standard statistics problem of estimating a quantile. Once we decide how wide a confidence interval we can tolerate for the quantile and what confidence we want that the quantile is in that interval, we can compute how many samples we need to take, using standard statistical techniques.[14]

We can compute the threshold offline. If we wish, we can use an anytime algorithm, running more samples over time so that our threshold will get better and better.

15. WORKSHOP DISCUSSION

Some highlights of the discussion of this paper at the 1998 New Security Paradigms Workshop follow.

Q: A core premise of the paper is that people have good common sense about independence. Granting that humans have good common sense about independence of humans, do they have good common sense about independence of computer artifacts? I doubt it.

A: No, we agree. Because computer artifacts are complex and because the interdependencies among them are partly hidden, humans probably lack good common sense about them. However, we claim that the kinds of factors that need to be considered when reasoning about the independence of humans and of computer artifacts are much the same. If that is not true, then this paper has no value.

Q: Systems employing Byzantine algorithms often do not withstand malicious attack. They may use unprotected communication, for example.

A: Byzantine algorithms *can* be used to achieve compromise tolerance, but are not enough. The Related Work section has been revised.

Q: Can the model deal with *unintentional* compromises, such as leaving the key to the machine room at a restaurant?

A: Yes. One can create an influence representing such an event – an unintended exposure of the machine room, for example – estimate its probability, and predict which agents would be compromised as a result.

Q: Can the model handle attempts to compromise agents by deceiving them – by messing with their inputs? If the agents are intrusion detectors, this is a very plausible mode of attack.

A: Yes. If two agents are likely to fall for the same deception, one can model that threat as a vulnerability affecting both agents. In the case of intrusion detectors, this event is more probable if the two detectors use the same analytical method, or if the two detectors employ many of the same sources (directly or indirectly). Thus the model can help ensure that a distributed intrusion detection system always uses more than one analytical method to confirm an attack, and that the system avoids double counting of its sources.

Q: Removing a compromised operator does not necessarily remove the compromise.

A: Certainly true. The model we gave does not represent time. Without changing the model, one could let some influence arcs represent past influences whose effects linger. By changing the model, one could add some representation of history.

Q: Does the model include a measure of how sure I am that I have enumerated all of the influences on a given agent?

A: No single number in the model captures this. However, if you know a lot about what influences a given agent, you put large values in its row and column of the barrier matrix (see Section 7.1), and otherwise you put small values.

Q: Could you add confidence factors, to deal with the fact that the probability estimates may be off?

A: Yes, that could be a good enhancement. There has been work on hedging probabilities, such as Dempster-Shafer theory. We have added this to the Enhancements section. But first, we should try a sensitivity analysis. If the results of the model are not too sensitive to the exact probability values, then we can relax.

Q: The weights in the strength-of-influence matrix may be hard for humans to estimate, but may be easy to measure. One could observe the "peacetime" behavior and see what influences whom.

A: This is appealing: an additional source of ground truth. There are learning algorithms for Bayesian networks.

Q: This model is reactionary. It discounts lone prophets. By the time their reports are corroborated, it may be too late!

A: Yes. That is a limitation. We suspect it may be fundamental. How can we ask a computer program to make an independent judgement about whether an information source is sincere? That seems to require human judgement.

Q: How would you attack a system based on the model?

Q2: You would find a least-cost spanning tree of agents and trick those (or otherwise compromise them). How much additional work does this impose on attackers?

A: The model has a tunable *threshold* that allows one to tune how much work the adversary will have to do. Of course, it also affects the rate of false positives and false negatives.

The model should come with a measure of the work imposed on attackers. This would have to involve a measure of the cost of exploiting each vulnerability. We have added this to the Enhancements section.

16. CONCLUSIONS

A designer can eliminate many single points of vulnerability in a distributed system by requiring corroboration from independent principals, or agents. For corroboration to be meaningful, each agent making decisions must have a system of judging which principals are likely to be independent. Dynamic use of this system can be viewed as diagnosis, because the analyzer tries to account for the news it receives, deciding whether to believe the news or to believe that the sources of the news lie. Thus, a strong hope for making a system tolerant of compromise is to integrate continual diagnosis into each agent's moment-by-moment decision making.

For agents to make independence judgments, they must have sources of information bearing on independence. Administrators who set up and maintain an authentication infrastructure can supply independence information as they introduce principals. The cost of doing so appears to be bearable. A great deal of independence information can be extracted from existing authentication infrastructure.

Though administrators can supply independence information only about first-hand, manually introduced principals, it appears that it will be feasible to infer independence nth-hand, and therefore to leverage existing independence information.

These enhancements will make authentication infrastructures far more tolerant of compromise, and therefore more survivable.

17. ACKNOWLEDGEMENTS

This work was funded by the Information Survivability program of the Defense Advanced Research Projects Agency (DARPA).

The paper benefited greatly from comments by and discussions with Don Bolinger, Mary Ellen Zurko, David Black, Rich Simon, Tom Sanfilippo, and others.

18. REFERENCES

[1] Abdul-Rahman, A. and Halles, S. A distributed trust model. Proceedings of the New Security Paradigms Workshop (1997).

[2] Brewer, D.F.C. and Nash, M.J. The Chinese Wall security policy. Proceedings of the Symposium on Research in Security and Privacy (May 1989), 206-214.

[3] Charniak, E. Bayesian Networks without Tears. AI Magazine, 12(4) (1991).

[4] Friedman, N. and Halpern, J.Y. Plausibility measures: A user's guide. Proc. Eleventh Conf. On Uncertainty in Artificial Intelligence (1995).

[5] Garfinkel, S. PGP: Pretty Good Privacy. O'Reilly & Associates, 1995.

[6] Gong, L., Lincoln, P., and Rushby, J. Byzantine Agreement with authentication: observations and applications in tolerating hybrid and link faults. Proceedings of the IFIP Working Conference on Dependable Computing for Critical Application, 5 (September 1995), 139-157.

[7] Heckerman, D., Geiger, D., and Chickering, D.M. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, v. 20, 1995

[8] Lyn, M.R., ed. Handbook of Software Reliability Engineering (McGraw Hill Text, April 1996).

[9] Malkhi, D. and Reiter, M. Byzantine quorum systems. Proc. 29th ACM Symposium on Theory of Computing (May 1997), 569-578.

[10] Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, 1988, 1991 (2nd print).

[11] Reiter, M.K. and Stubblebine, S.G. Toward acceptable metrics of authentication. Proc. 1997 IEEE Symposium on Security and Privacy (May 1997), 10-20.

[12] Reiter, M.K. and Stubblebine, S.G. Path independence for authentication in large-scale systems. Proceedings of the 4th ACM Conference on Computer and Communications Security (April 1997), 57-66.

[13] Rice, J. Mathematical Statistics and Data Analysis. Wadsworth, Inc., 1988. 191.

[14] Rice, J. Op. Cit. 342.

[15] Rosenberry W., Kenney D., and Fisher, G. Understanding DCE. O'Reilly & Associates, Inc., 1992.

[16] Simon, R. and Zurko, M. Separation of duty in role-based environments. Proc. Computer Security Foundations Workshop (September 1996).

[17] WWWebster Dictionary (1997, Merriam-Webster). http://www.m-w.com/cgi-bin/dictionary

[18] Zurko, M.E. and Simon, R.T. User centered security. Proc. New Security Paradigms Workshop (September 1996).