# Using economics to model threats and security in distributed computing
# (Extended Abstract)

Yvo Desmedt, Mike Burmester
Department of Computer Science
Florida State University
Tallahassee, FL 32306-4530, USA
{desmedt,burmester}@cs.fsu.edu

Yongge Wang
Karthika Technologies Inc.
Canada
ywang@karthika.com

**Abstract**

The traditional approach to address security threats in distributed computations is based on the Byzantine faults model. We argue that this approach is inappropriate for current applications in which the adversary can exploit the homogeneity and symmetry of the model and propose a new model which are based on principles from economics, game theory, and combine it with artificial intelligence and control theory.

## 1 Background

The traditional Byzantine model was introduced more than 20 years ago to deal with faults in computer systems and has been used extensively for secure distributed computation. This model is homogenous, in the sense that it makes no distinction between the nodes (computers) that the adversary can attack. The adversary is only bounded by the number of nodes that can be attacked. We argue that this model is inadequate for current applications.

We now extend the Byzantine faults model.

## 2 Using economics to express feasible attacks

We shall assume the adversary $A$ has a *budget* $B_A$ (not necessarily monetary), which is a bound on the available attack resources. In the case of a hacker the budget could be expressed in the number of leisure hours.

Let $G = (V, E)$ be the computer network, with $V$ the set of nodes (computers) and $E$ the set of links (edges). To each subset $S$ of $V \cup E$ we assign a cost $c_{S,A}$. The adversary is restricted to attacks on those sets $S$ for which $c_{S,A} \leq B_A$. This induces a structure $\Gamma_G$ on the subsets $S$ of $V \cup E$ which we call the *attack access structure* of the adversary $A$.

This quite general model can be used as basis for alternatives to the Byzantine model. However, since the number of subsets $S$ may be exponential (in the size of $G$), it is impractical to assign explicitly cost values $c_{S,A}$ when $G$ is large. Therefore it may be necessary to use an implicit approach. However, any such such approach must be *stable* in the sense that the cost assignments of the implicit model should not be significantly different from those of the general model in real world cases.

### 2.1 Examples of implicit cost measures

Let us first focus on a simplified model in which the links of the computer network $G$ are ignored.

A first approach is one which the cost assignment function is *linear* in the number of computers. In this case, the cost of attacking any machine is uniform, and the cost of attacking any $k$ machines is $k$ times this basic cost. Obviously, this model is not realistic. Indeed, the cost of attacking two computers running on the same platform (e.g. same operating system) is not twice the cost of attacking a single computer.

A more realistic approach is to set the cost of attacking *any* number of computers running on the same platform to be the same as that of attacking a single computer on that platform. We call this, a *platform oriented* model. Such a model could be used for networks in which several computers use the same operating system. This model has several variations. For example, the cost to undermine an operating system could be system dependent. Indeed the cost of breaking into an operating system designed with state of the art security may be much higher than that of breaking into a system believed to be less secure. Alternatively, the cost of breaking into two or more systems may be additive (equal to the sum of the costs of breaking into each individual system). Other versions can be strongly non-linear.

One can obviously make several variants of this platform oriented model. For example one could assign a basic cost for breaking into an operating system and then add an extra cost per machine to break into. However, when automated attacks (such as viruses, worms and their hybrids) are used, this extra cost may be negligible.

It is interesting to reflect back on the typical threshold approach used in several papers on secure distributed computers. In this model the attack access structure $\Gamma_G$ consists of all sets $S$ that have at most $k$ computers. More specifically:

- for each subset $S \in \Gamma_G$ of at most $k$ computers the cost $c_{S,A}$ to attack those computers is less than the budget $B_A$ of the enemy,
- for each subset $S' \notin \Gamma_G$ of $k+1$ computers, or more, the cost $c_{S',A}$ is larger than $B_A$.

We call these conditions the *Byzantine threshold cost assumption.*

Let us now include in our discussion the links of the computer network $G$. If the enemy has control of a node, it evidently also has control of its links. So, what remains to be discussed is an attack against a subset $V'$ of nodes and a subset $E'$ of non-adjacent links. This again introduces a non-linear cost effect. The enemy could attack these links physically. Note that the cost for the enemy to attack a remote link may be substantially higher than that to attack a nearby one. This explains why our cost model is dependant on the enemy. Note that if the cost to attack non-adjacent links is prohibitive, the node-only based approach is appropriate. The idea of using link encryption can be viewed as based on an economic model in which the cost to attack nodes is prohibitive.

### 2.2 Generalizations

With broadcast technology (such as ethernet), it is more natural to use hypergraphs. In this case the edges are replaced by hyperedges.

## 3 Economics of the adversary

Choosing the largest subset of $\Gamma_G$ may be appropriate for hackers, but it will not necessarily cause the most damage. A number of issues affect the optimality of an attack. Here we discuss one such issue.

For each (network) application $a$, a sub-network consisting of several computers and links $T_a \subset V \cup E$ may be involved. We associate to each application involving $T_a$, a network *flow* $f_{T_a}$. The maximal flow can be expressed as the capacity $C_{T_a}$.

Depending on the application, the economic importance of flows may be different. We therefore assign to each flow $f_{T_a}$, an *economic impact factor* $I_a$. The *total impact of the application* is then $f_{T_a} * I_a$. When considering the economic impact of a computer network as a whole, one cannot just add these separate total impacts since flows are not necessarily additive. Indeed some flows may be linked to several applications. Moreover, as pointed out by Martelli and Montanari, data can be copied in nodes and so data flows do not follow the laws of fluid dynamics. Therefore when looking at the economic impact of a system as a whole, we have to deal with a *weighted* total flow $F$ and a *weighted* total capacity $C$.

What the adversary now will do is to choose a subset $S \in \Gamma_G$ to attack. That will reduce the weighted total capacity of the system from $C$ to $C_{\bar{S}}$. $C_{\bar{S}}$ is the remaining weigthed total capacity that can be achieved by the remaining of the networked computers, expressed as $\bar{S} = (V \cup E) \setminus S$. The enemy can optimize the attack by choosing an $S \in \Gamma_G$ for which $C_{\bar{S}}$ is minimal. If $C_{\bar{S}}$ is below a certain critical minimal value $C_{crit}$, the enemy has won. We call $C_{crit}$, the *critical* capacity of the computer network. If the enemy cannot win, the best strategy for the enemy is to optimize the attack by choosing an $S \in \Gamma_G$ with minimal capacity $C_{\bar{S}}$. Note that the remaining capacity will be different depending whether the enemy destroys nodes or takes control of nodes (in a Byzantine attack).

### 3.1 Generalizations

Clearly the assumption that critical capacity $C_{crit}$ is static is restrictive. In a more realistic model one has to allow for dynamic factors influencing the value of $C_{crit}$.

### 3.2 Analysis of the extended model

To illustrate this we consider the typical Byzantine model in computer networks. With the Byzantine model we have symmetry and linearity: for each application, a successful attack has the same economic impact factor and the cost function is linear.

Now suppose that the computer network is $n$-connected, and that we are only concerned with reliability (and not privacy or integrity). Let the impact factor be 1 and the minimal capacity over all the concerned disjoint paths be $c$ bits/sec. Then the optimal strategy for the enemy is to take over a set $S$ of $\lceil n/2 \rceil$ nodes on disjoint components of the communication/computation of $G$ (assuming the access structure $\Gamma_G$ allows for such an attack) and send erroneous data. This will reduce the capacity of the computer network to 0.

# 4 The economics of the designer

The goal of the designer is to design a network $G = (V, E)$ which is such that the cost of a successful attack is more than the resource available to the adversary. The designer's strategy is determined by (at least): a design budget $B_D$, the required (weighted) flow capacity $C_D$, the economic impact factor threshold $C_T$ and the adversary's budget $B_A$.

The designer wins if the network $G$ is such that: $(i)$ the cost of the system is no more than the budget $B_D$, $(ii)$ the total flow $f_{V \cup E}$ is no more that the capacity $C_D$, and $(iii)$ the enemy cannot win.

# 5 Extensions

We briefly describe two extensions.

## 5.1 Using AND/OR networks for distributed computing

The Byzantine based model is inadequate for modeling real life distributed computing. With distributed computing several operations may take place and the computation is only completed when *all* operations have been performed. Such systems can be modeled by PERT (Program Evaluation and Review Technique) networks. However PERT networks cannot model redundancy, so we use AND/OR networks.

These also allow to marry the computer and mechanical worlds.

## 5.2 Control theory variants

We must also take into account *time aspects*. The timing aspects of an attack involve several factors such as buffers, the time between the attack and its detection of the attack, the recovery time, etc. The designer wins if: (time to repair the system) + (time to detect attack) $\leq$ (time of no return) + (shelf-time of the stock).

# 6 Discussion

The Byzantine model has been used as the main approach to model secure distributed computation. Although it has lead to interesting theoretical results, it cannot deal with the impact on information security of a homogeneous (monopolistic) platform in which the cost to attack extra computers running the same platform is minimal. Our economics model allows to deal with such issues.