

# A UNIFIED FRAMEWORK OF INFORMATION ASSURANCE FOR THE DESIGN AND ANALYSIS OF SECURITY ALGORITHMS

Alvaro A. Cárdenas\*, Gelareh Taban and John S. Baras  
Electrical and Computer Engineering Department and the Institute for Systems Research,  
University of Maryland, College Park, MD, 20742

## ABSTRACT

Most information security algorithms cannot achieve perfect security without incurring severe operational costs such as false alarms, network congestion, capital investment etc. Operating or designing an algorithm with perfect security is therefore not an economically rational alternative and thus the question arises of how to find the appropriate tradeoff between security and its costs. Although several other researchers have recognized that there is a tradeoff, there is very little work in formally characterizing it. This paper provides the first steps towards a more systematic and general approach for cost-effective security management.

## 1. INTRODUCTION

In order to provide a formal guarantee of security, the algorithms used to ensure several information security goals, such as authentication, integrity and secrecy, have often been designed and analyzed with the help of formal mathematical models.

One of the most successful examples is the use of theoretical cryptography for encryption, integrity and authentication. By assuming that some basic primitives hold, such as the hardness of factoring large composite numbers (the problem RSA algorithms relies on), or the hardness of computing discrete logarithms (the problem elliptic curve cryptography relies on), some cryptographic algorithms can formally be proven secure.

Formal security models however have theoretical limits, since it cannot always be proven that an algorithm satisfies (or not) certain security conditions. For example, as formalized by Harrison, Ruzzo, and Ullman, the access matrix model (the determination of whether or not a given subject can ever acquire access to a given object) is undecidable, Rice's theorem implies that static analysis problems are also undecidable (i.e. finding bugs in computer programs can never be done with perfect accuracy), Fred Cohen proved that it was impossible to build an anti-viral tool that would detect all possible computer viruses, and

using a similar argument, it can be inferred that most intrusion detection problems are undecidable as well.

Besides computational intractability of several problems, there are several other problems with inherent uncertainties, such as spam detection, fingerprinting of multimedia data and MAC layer misbehavior.

Any algorithm trying to solve these hard or undecidable problems is bound to produce a non-negligible amount of decision errors. Therefore we cannot achieve security in the traditional computational model.

The evaluation of the performance of these imperfect algorithms is a very important part of information assurance, yet we do not have a framework that provides a sound analysis of these security solutions. Notice however that all algorithms trying to solve these "impossible" problems are forced to make approximations and can consequently be formulated as a tradeoff between the costs of the security algorithms (i.e., the necessary resources for their operation) and the correctness of their output (i.e., the security level achieved). In practice it is very difficult to assess both: the real costs of these security solutions and their actual security guarantees. Most of the research therefore relies on ad hoc solutions and heuristics that cannot be shared between security fields trying to address similar problems. It is our goal therefore in this paper to provide a more robust and systematic characterization of the tradeoffs between security and the costs associated with a given security technology.

The main components of any security evaluation are the use of security metrics and the adversary model. In this paper due to space constraints we focus only on the evaluation metrics, leaving the detailed adversarial model for other work. Our focus is to design metrics that aid the decision-making process in determining the optimal level of security investments and the optimal allocation of the resources available. Another objective is to use metrics that are flexible enough to accommodate different resources such as monetary costs (operating costs and capital investments required by the security solution), network efficiency costs (e.g. network congestion used by the security solution), the

number of false positives, the time required to obtain a solution, etc.

We focus on two problems in this paper. The first problem is in the evaluation of intrusion detection systems. Intrusion detection systems are essential tools for detecting any breaches of security in computer networks. We evaluate them by using security metrics based on the class imbalance problem (i.e., incorporating the fact that normal events occur much more frequently than attack events) and risk management techniques.

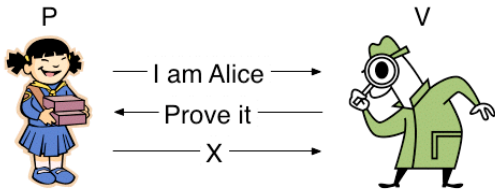
The second is an example for sensor networks. Sensor networks are a key element for future battlefield scenarios, since they can provide a wide variety of information about a region of interest. Sensor nodes however can be tampered or even compromised, and as a result can give erroneous information that can make the final user of the data to make incorrect decisions. We therefore compare and analyze different alternatives to the data aggregation problem emphasizing again the evaluation metric used.

## 2. METRICS FOR DETECTION ALGORITHMS

### 2.1 Motivation

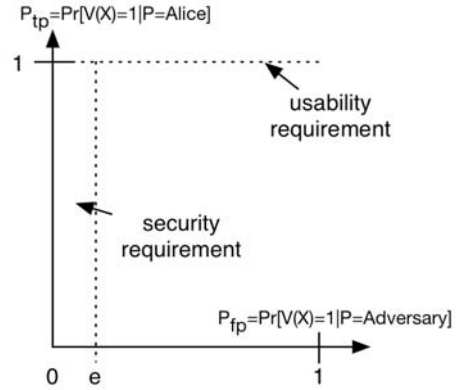
Several algorithms for information assurance such as intrusion detection systems (IDSs), static analysis tools and anti virus software can be modeled as detection algorithms. Their task is to raise an alarm whenever there is an attack (intrusion/software bug/virus). Despite the prevalence and usefulness of these algorithms, there is no useful metric so far to evaluate their performance, optimize their configuration and allow for the comparison among different detection alternatives.

Other security algorithms on the other hand, in particular cryptographic algorithms, can also be modeled as detection algorithms with well defined evaluation metrics. It is insightful for our study to first analyze the performance metrics of these cryptographic algorithms and why they are not enough for our case.



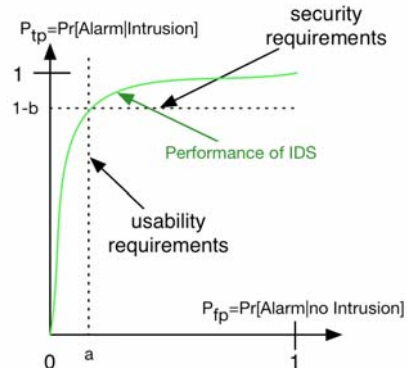
**Fig. 1** Authentication algorithm V should output 1 if and only if P is who she claims she is.

Consider the public-key authentication algorithm given in Fig 1. In this protocol P needs to convince the verifier V that it has a secret only Alice can know (the secret-key of Alice). The verifier V can be seen as a detection algorithm, which outputs  $V=1$  if it believes P is Alice and  $V=0$  otherwise.



**Fig. 2** Evaluation of the correctness and security of an authentication algorithm.

The formal evaluation metrics for this algorithm are shown in Fig 2. In particular there is a **usability metric** and a **security metric**. The usability metric measures the correctness of the algorithm; mainly that Alice should always be authenticated. Formally, the probability of a true positive should be one:  $\Pr[V(X)=1|P=Alice]=1$ . The security metric on the other hand requires that the probability of authenticating an impostor is upper bounded by a very small quantity. Formally, the probability of a false positive should be less than  $e$  ( $\Pr[V(X)=1|P=Adversary]<e$ ). Usually  $e$  is a function that decreases as  $O(2^{-k})$  where  $k$  is the length of the secret-key. Therefore by having a “large” secret key (e.g.  $k=128$ ) the probability that the adversary is authenticated is negligible.



**Fig. 3** Our problem: we can no longer achieve negligible error probabilities, i.e.  $a$  cannot be made as small as possible without increasing  $b$ .

Our problem on the other hand is that most of the algorithms we are considering are undecidable, and thus

we cannot achieve usability and security requirements without incurring large tradeoffs between the two. Fig 3 shows a typical example of the problems we face. The green line represents the possible operating points of an IDS and it is usually called the **ROC curve**. Notice that we can achieve perfect usability with  $a=0$  (no false alarms) by not using the system:  $b=1$  (never detect intrusions). Similarly perfect security is achieved for  $b=0$  (all intrusions are detected) at the cost of having false alarms all the time ( $a=1$ ). Obviously operating with perfect security is not economically rational in this case. The two basic questions for the evaluation of detection algorithms are therefore, what is the best tradeoff between  $a$  and  $b$ ? And given fixed misclassification probabilities  $a$  and  $b$ , are they good enough?

## 2.2 The Building Blocks

Let  $I$  be an indicator random variable denoting if there is an attack ( $I=1$ ) or not ( $I=0$ ), and let  $A$  be the indicator random variable of the output of the detection algorithm: alarm ( $A=1$ ) or normal event ( $A=0$ ). As we have seen in the previous section, the most basic metrics are the **probability of false alarm** (false positive rate)  $P_{FA}=\Pr[A=1|I=0]$  and the **probability of detection** (true positive rate)  $P_D=\Pr[A=1|I=1]$ . These two values are in fact the basic building blocks for several evaluation metrics that have been proposed in several fields, including machine learning, detection theory, cryptography, medical test diagnosis, and even computer security.

For example  $P_D$  is also known as the **sensitivity** of the detection test for medical diagnosis, or as **recall** in the information retrieval literature. Similarly  $P_{FA}$  is also known as the **specificity** of a test.

The problem with this multidisciplinary approach for detection metrics is that they are introduced with different names and assumptions, and this leads to the reinvention of the metrics and sometimes confusion on how to interpret the metrics. In particular we point out that several of these metrics are not useful and in fact are sometimes misleading for the problems that we face in computer security. In the following we define the previously proposed metrics in a unified way and try to address their advantages and disadvantages.

## 2.3 The Base-Rate

The **base-rate** is not a metric per se, but yet another building block for several other metrics. The base-rate for our case is defined as the likelihood of an attack, i.e.  $p=\Pr[I=1]$ . This value is also known as the **prevalence** in medical diagnosis or as the **prior** in

Bayesian decision making. The problem with prevalence in computer security is its highly unpredictable nature. If we try to estimate  $p$  different times we will get very different values. It is also important to note that for several computer security problems, the amount of normal events outnumber by a very large value the amount of attack events, and thus  $p$  tends to be smaller than in any other field, e.g.  $p=10^{-5}$ . This presents unique challenges for the design and evaluation of information assurance algorithms.

## 2.4 Accuracy

Probably the most widely used metric in machine learning is the **accuracy** of the classifier, or the **probability of correct classification**, given by  $\Pr[A=I]=(1-P_{FA})(1-p)+P_Dp$ . The problem is that for small  $p$ 's the accuracy is close to  $(1-P_{FA})$ , thus disregarding  $P_D$  completely. The same is true for the **probability of error**:  $1-\Pr[A=I]$ .

## 2.4 The Positive Predictive Value

Suppose for example that you get an alarm in a test with large  $P_D$  and very small  $P_{FA}$ . The naïve conclusion is that this alarm is very likely to be due to an attack. However this might not be true for small base-rates.

As the **positive predictive value (PPV)** (also known as **precision** in the information retrieval community or as the **Bayesian detection rate** in the IDS literature) shows us, if we compute the **posterior** probability of an intrusion given an alarm we obtain the following Eq.:

$$PPV = \Pr[I = 1 | A = 1] = \frac{P_D p}{P_{FA}(1-p) + P_D p} \quad (1)$$

So if  $P_D=1$ ,  $P_{FA}=0.01$  and  $p=10^{-5}$  then  $PPV=0.001$  (i.e. of 1000 alarms, on average only one is due to a real attack). This problem is due to the fact that it is difficult to interpret what a small false alarm rate is when  $p$  is also small. In fact we believe that several detection algorithms “get away” by presenting results with apparently low  $P_{FA}$  without taking into consideration the real impact of the false alarms. Since one of our main goals is to prevent the misinterpretation of the metrics, we believe that a good estimate for a low false alarm rate is to set  $P_{FA} \approx p$ , since in this case  $PPV \approx 0.5$  (assuming  $P_D$  is high). In general we can define a new evaluation procedure as follows:

$$\begin{aligned} & \max_{(P_{FA}, P_D) \in ROC} P_D \\ & \text{subject to: } PPV \geq c \end{aligned} \quad (2)$$

Another metric that has been proposed is the use of the **negative predictive value (NPV)**  $=\Pr[I=0|A=0]$ . As

we showed in (Cárdenas et al. 2006) this metric is not very relevant for small values of  $p$ .

## 2.5 F-Score

The F-score, or F-measure, is a metric used by the information retrieval community. We believe it is very valuable for our case since it combines precisely the two metrics that we are most interested in: PPV and  $P_D$ . We are not aware of any other widespread used metric focusing only in this particular tradeoff. The F-score is defined as

$$F - score = \frac{(\beta^2 + 1)PPV \times P_D}{\beta^2 PPV + P_D} \quad (3)$$

when  $\beta=1$  the F-score is balanced. It favors PPV when  $\beta>1$  and  $P_D$  otherwise.

## 2.5 Tradeoff Curves

The only problem we see with the PPV value is its dependence on a reliable estimate of  $p$ . We therefore introduced the **IDOC** curves (later renamed to **B-ROC** curves) in (Cárdenas et al. 2006) as the tradeoff between the  $P_D$  and 1-PPV for a range of uncertain values of  $p$ . This characteristic, the fact that we disassociate  $p$  from the computation of the B-ROC curves, is what makes these curves different from the similar **Precision and Recall** curves used by the information retrieval community; since these last curves are computed in a dataset with a given  $p$  (i.e. they are always dependent on a given  $p$ ).

We believe that B-ROC curves are a better alternative to ROC curves as they provide an easier interpretation of the results, and to Precision and Recall because of their independence from  $p$ .

Another alternative to ROC curves are the **Pseudo-ROC** curves, in which the x-axis is not  $P_{FA}$  but the **incidence** of the false alarms, which is the number of alarms per unit of time as opposed to number of alarms per event being classified. For example if the unit of time is “day” then the x-axis is  $P_{FA}N/day$ , where N is the number of normal events monitored per day. This prevents the misinterpretation of small  $P_{FA}$ . These curves were introduced by the speech processing community, and they were used to evaluate the IDSs participating in the DARPA-MIT-Lincoln Labs competition.

Finally, a very popular metric used in machine learning and medical tests is to take the area under the ROC curve (AUC). Again we argue that this metric is misleading for small  $p$  values as classifiers with very

different performance can have exactly the same AUC. A possible solution is to consider only the area under the ROC curve until  $P_{FA}=p$ . Then again this is only a heuristic.

## 2.5 Risk Metrics

Risk is essentially the combination of the probability of an event and its consequence, and risk management is the process of shifting the odds in your favor by finding among all possible alternatives, the one that minimizes the impact of uncertain events.

Probably the best well known risk metric is **the average loss**

$$R_\mu = E[L] = \sum_{i=1}^n L_i p_i \quad (4)$$

Where  $L_i$  is the loss if event  $i$  occurs, and  $p_i$  is the probability that event  $i$  occurs.

In the finance literature this metric is usually called the **Annual Expected Loss (ALE)** because each period corresponds to a fiscal year, and therefore  $L_i$  corresponds to *annual* dollar losses if event  $i$  occurs. Similarly  $p_i$  corresponds to the probability of event  $i$  occurring in a given year. The ALE can help in the decision process of which security technology should a company invest in.

This same risk metric was used in (Gaffney and Ulvila 2001) for the evaluation of Intrusion Detection Systems. In this case  $L_{01}$  corresponds to the cost of responding to a false alarm, and  $L_{10}$  corresponds to the cost of failing to detect an intrusion. The average loss (per event being monitored, as opposed to average loss per year) then becomes

$$R_\mu = L_{01} \Pr[I = 0, A = 1] + L_{10} \Pr[I = 1, A = 0] \quad (5)$$

a quantity that can be interpreted with some of our previously defined metrics, such as the probability of false alarm and the probability of detection or the PPV value and the probability of detection:

$$\begin{aligned} R_\mu &= L_{10} P_{FA} (1 - p) + L_{01} (1 - P_D) p \\ &= L_{10} PPV \Pr[A = 1] + L_{01} (1 - P_D) p \end{aligned} \quad (6)$$

This metric not only helps us in deciding among different IDS alternatives, but also in the configuration of a given IDS to its optimal operating point in the ROC. If we let  $x = P_{FA}$ , then ROC can be seen as a function  $f()$  where  $f(x)$  is the corresponding probability of detection for a given probability of false alarm  $x$ . With this notation, the minimization of the average loss then becomes:

$$\min_x R_\mu(x) = \min_x L_{10} x(1 - p) + L_{01} (1 - f(x)) p$$

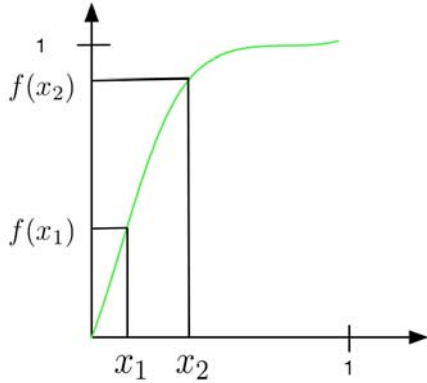
Taking the derivative and equating it to zero, we obtain the condition of optimality for  $x^*$ :

$$f'(x^*) = \frac{1-p}{p} \frac{L_{10}}{L_{01}} \quad (7)$$

That is, the optimal tradeoff between  $P_{FA}$  and  $P_D$  is at the point where the slope of the ROC  $f'(x)$  equals the value provided in Eq. (7).

The strict tradeoff between the probability of false alarm and the probability of missing an intrusion given by the ROC curve might be very limiting in several practical scenarios. Sometimes it is also difficult to force the detector to make a hard decision, in particular in cases where not enough evidence is given in order to classify an event.

We therefore now introduce a new interpretation of the average loss metric. In this new setting we define a new possible decision for the IDS:  $A = s$ . The motivation for introducing this kind of output is for being able to label certain network events as “suspicious” while not raising an alarm.



**Fig. 4** The indecision region is determined by the points  $x_1$  and  $x_2$  and their corresponding ROC values  $f(x_1)$  and  $f(x_2)$ .

With this new output the average loss can now be defined with the help of Fig. 4.

$$R_\mu = L_{01}x_1(1-p) + L_{0s}(x_2 - x_1)(1-p) + L_{10}(1-f(x_2))p + L_{1s}(f(x_2) - f(x_1))p \quad (8)$$

taking the gradient of  $R_\mu$  and equating it to zero we obtain the following conditions:

$$f'(x_1^*) = \frac{1-p}{p} \frac{L_{01} - L_{0s}}{L_{1s}} \quad (9)$$

$$f'(x_2^*) = \frac{1-p}{p} \frac{L_{0s}}{L_{10} - L_{1s}}$$

Notice that the above conditions only hold if

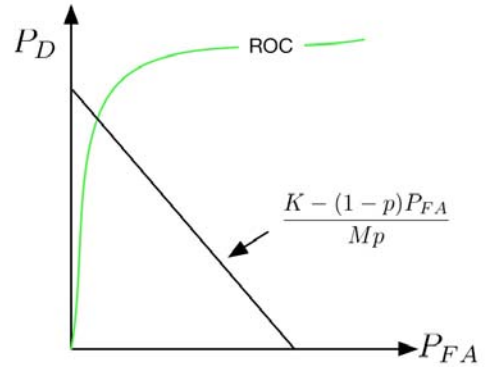
$$\frac{L_{0s}}{L_{10} - L_{1s}} \leq \frac{L_{01}}{L_{10}} \leq \frac{L_{01} - L_{0s}}{L_{1s}} \quad (10)$$

(If Eq. (10) does not hold then we should not use the output  $A = s$ .) A natural interpretation of these conditions is to assume that from all the suspicious events, the IDS operator is only going to check a percentage  $\gamma$  of them due to time and load constraints. Therefore  $L_{0s} = \gamma L_{01}$  and  $L_{1s} = \gamma L_{10}$ . With this values Eq. (10) implies that  $\gamma < 0.5$ . The interpretation of this result is that using a “suspicious” alert is only cost-effective when less than half of these suspicious reports are investigated by the IDS operator.

Others risk metrics try to get more information about the probability distribution of the losses, and not only its mean  $R_\mu$ . For example the variance of the losses

$$R_\sigma = E[L^2] - R_\mu^2 \quad (11)$$

is very useful in finance since it gives more information to risk averse individuals. This is particularly important if the average loss is computed for a large period of time (e.g. annually). If the loss is considered every time there is a computer event then we believe the average loss by itself provides enough risk information to make a rational decision.



**Fig. 5** The point in the ROC that maximizes  $P_D$  while keeping the number of alarms (per day) bounded by  $K$  can be found by the intersection of the ROC and the line of Eq.(14).

## 2.6 Workforce Utilization

Assume now that the operators of an IDS can only process a fixed number of alarms. In this case we want to find the point in the ROC that will give on average a number of alarms bounded by  $K$  over a specific amount of time (e.g. one day):

$$E[N] \leq K \quad (12)$$

where  $N$  is the number of alarms.

Before being able to compute  $E[N]$  we need to have an estimate of the number of network events  $M$  per unit of time (e.g. number of alarms per day). With this estimate  $N$  is then a binomial distribution of  $M$  trials and with probability of success  $\Pr[A=1]$ . Therefore

$$E[N] = M \Pr[A = 1] = M (P_{FA}(1-p) + P_D p) \quad (13)$$

Solving for  $P_D$  in Eq. (12) we have

$$P_D \leq \frac{K}{Mp} - \frac{1-p}{p} P_{FA} \quad (14)$$

If we want to maximize also  $P_D$  then we would find the point in the ROC curve that satisfies equality in Eq. (14). Fig. 5 shows how to find the optimal operating point of the ROC satisfying the workforce utilization constraint.

We believe this metric is very useful again for avoiding misinterpretation of the real impact of the false alarms.

## 2.7 Summary

We have introduced in this section several widely used metrics for the evaluation of detection schemes and discussed their advantages or disadvantages for the evaluation of detection schemes, in particular focusing on IDSs. In some frameworks such as the Risk metrics we also proposed new paradigms that we believe are useful for the evaluation of IDSs. In future work we plan to test the effectiveness of these metrics in realistic case examples.

We believe that Risk metrics, B-ROC curves, the F-Score, the Workforce Utilization and the Pseudo-ROC curves are the most meaningful for our problem. It is difficult to decide which metric should be used all the time; instead these metrics provide a complimentary view of the detection algorithm.

## 3. SENSOR NETWORK AGGREGATION

In recent years, there has been much interest in the area of sensor networks. An emerging problem is that due to a lack of a unified framework, it is difficult to compare proposed algorithms. In this section, we concentrate on the problem of *secure data aggregation* in sensor networks and propose metrics that allow network administrators to quantify the security of each algorithm for comparison.

Current aggregation solutions deal with two main problems. The first problem, which we will refer to as *network aggregation*, deals with aggregation of sensed

readings in a network where an adversary can compromise one or more nodes. The administrator's objective is to minimize the amount of damage that an adversary can inflict while minimizing the congestion in the network. The second problem, referred to as *function resiliency*, deals with the resiliency of the aggregation function to misbehavior. The administrator wants to ensure that the aggregate that is computed is correct given a set of sensed readings, while minimizing the required resources.

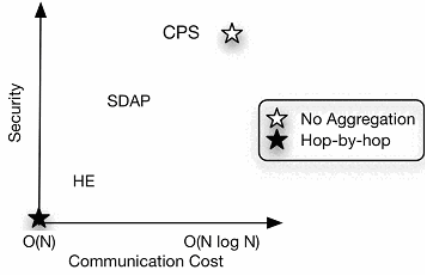
Network aggregation considers the model where a base station (BS) sends a query to the network and the nodes in the network respond with their appropriate readings. To minimize network congestion, nodes perform hop-by-hop aggregation, whereby routing nodes aggregate their values with the value being routed. The total congestion and the edge congestion are optimal at  $O(N)$  and  $O(1)$  respectively, where  $N$  hereafter, is the total number of sensors in the network. Assuming an authentication primitive, e.g. message authentication codes (MAC), the scheme is secure against an outsider adversary (who does not have any authentication keys). However, by compromising a node, an adversary can modify not only its own reading but also the aggregate which represents the readings of all the sensors in the subtree of that node. Although this scheme has optimal communication overhead, it allows maximal damage by an adversary that compromises a single node (when the adversary compromises a node close to the BS).

On the other hand, minimum damage due to a single compromise can be achieved if the network does not perform any aggregation and each sensor individually sends their readings to the BS. Although a compromised node can at best modify only its own reading, a communication overhead of at best  $O(N \log N)$  and at worst,  $O(N^2)$  is incurred (depending on the routing structure).

Fig. 6 shows the design space of network aggregation algorithms with the bounds asserted by the hop-by-hop and no aggregation schemes. Fig. 6 also compares the security of some of the existing systems with the total congestion they incur. It is of interest to design a system that achieves the security of the no aggregation scheme with a communication overhead approaching that of the hop-by-hop scheme.

The second problem of aggregation is function resiliency, which assumes a single aggregator model. In this model, an aggregating node collects all the raw sensor readings and submits the aggregate to the BS. To minimize the probability of misbehavior by the aggregate, the BS can request all the sensor readings.

However this is in effect identical to the no aggregation scheme with similar cost. If the BS minimizes cost by requesting only the aggregate value however, it is most vulnerable to misbehavior. It is interesting to note that in this problem, communication between the aggregate and the BS does not need to be the only cost the administrator tries to minimize, as in SIA (Przydatek et al., 2003). The network administrator can also use algorithms where aggregate witnesses (Du et al., 2003) increase the confidence of the BS in the final aggregate.



**Fig. 6** Design space of aggregation algorithms HE (Hu et al, 2003), SDAP (Yang et al., 2006), CPS (Chan et al., 2006)

### 3.1 Evaluation Metrics

The major problem with existing solutions to aggregation is that there is no single formulation of a security metric and thus no common framework for comparison. In this section we first identify the weaknesses of existing security metrics and then propose new metrics that allow algorithms to be quantitatively compared.

In order to formalize an evaluation framework, we first need to introduce the adversary model in sensor networks. It is assumed that the adversary has a network-wide presence, can eavesdrop on all messages and can insert or modify messages at will. The adversary can be an outsider or an insider by compromising sensor nodes. Once a node is compromised, the adversary obtains control over the node’s secret data and subsequent behavior. In order to quantify the power of the adversary, we now define the following:

**Definition:**  $A(r,c)$  is an adversary that can affect the readings of  $r$  nodes that are not in its control and compromise  $c$  nodes.

Table 1 presents a comparison of network aggregation algorithms. HE measures security by the number of compromised nodes the algorithm is resilient to. This metric however does not account for how much damage an adversary can inflict if it compromises an extra node. CPS defines what they refer to as optimal

security but their metric cannot compare non-optimal algorithms. SDAP does not use a security metric.

In contrast to existing metrics, our metric tries to bound the damage of an adversary in network aggregation for different costs.

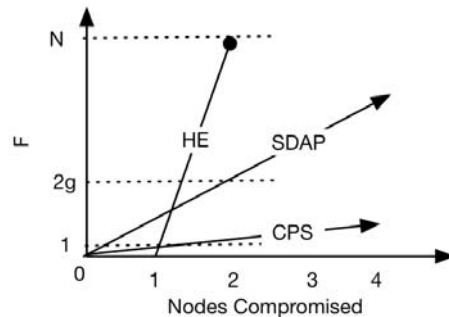
Define function  $F$ , such that given the set of values  $\{s_i\}_{i=1}^N$  of each node in the network, the aggregating structure used in the network  $T$ , and the adversary  $A(r,c)$ , outputs the number of node values that the adversary can affect. Therefore:

$$F(\{s_i\}_{i=1}^N, T, A(r,c)) \in [1, N] \quad (15)$$

Table 1 shows how this metric can be used to compare the security of network aggregation schemes: CPS achieves optimal security where  $F(.)=1$ , the security of SDAP depends on the size of its groups  $g$  and HE has least security with  $F(.)=N$  when the adversary compromises at least 2 nodes.

Another advantage of our metric is that it gives an intuition about how well the aggregation system degrades as the power of the adversary grows. Fig. 7 shows how well the system fares as the number of compromised nodes increases.

A unified security metric for aggregation function resiliency has also not yet been formulated. (Du et al., 2003) consider the number of witness aggregate nodes as a metric. (Mahimkar et al., 2004) use threshold cryptography to ensure sensors agree with the final aggregate. SIA measures its security by computing an approximate of sample values and bounding the distance of the approximation with the aggregate. Finally [Wagner, 2004] addresses some of the issues of measuring and bounding the contribution of an adversary to the final aggregate result.



**Fig. 7** Comparing how the security of different schemes degrades with the number of compromised nodes.

It is clear that the suggested metrics in the literature today are all very much bound to their algorithms. Our

approach however, bounds the damage caused by the adversary by relating the metric to a unified framework and the adversary.

Define function *Detect()* which uses a distribution  $D$  over the sensor readings, as well as the aggregate  $G$  and the adversary  $A(r,c)$ , to identify misbehavior by the aggregating node. The metric uses  $G^*$  as the ideal aggregate over the sensor readings.

**Definition:** An aggregation function is  $(\epsilon, \delta)$ -secure if the BS can successfully distinguish an adversary  $Adv(r,c)$  controlled aggregate that is within  $\epsilon$  of the ideal aggregate with probability  $\delta$ . Formally:

$$\Pr[Detect(|G - G^*| \geq \epsilon, D, Adv(r, c)) = 1] > \delta \quad (16)$$

This is a highly usable metric as it allows the network administrator to evaluate different aggregation functions based on the parameter important to them; for example maximizing probability of detection or minimizing the effect of the adversary on the aggregate. In the future, we plan to show how existing algorithms compare using our aggregate function resiliency metric.

## CONCLUSIONS

In this paper we have identified, improved and even proposed new metrics that take into account the tradeoff between the available resources and the security of an algorithm. We are currently working on the validation of these metrics in realistic intrusion detection and sensor network scenarios. We believe this new characterization of security is of critical importance for the sound design and evaluation of future security algorithms.

## ACKNOWLEDGMENTS

This work is prepared through collaborative participation in the Communications and Networks consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. Research is also supported by the U.S. Army Research Office under Award No. DAAD19-01-1-0494 to the University of Maryland, College Park.

Protocol	Security Metric	Security Claim	Total Congestion	Edge Congestion	$F(\{s_{i=1}^N, T, A(r, c)\})$
SDAP	None	BS detects any group abnormal aggregates and requests groups to attest. If attested, an adversary is detected with probability 1.	Depends on $g$ . For $g$ as large as $N$ , $O(N)$ . For small $g$ , $O(N \log N)$ .		$F = g$ , Where $g$ is the group size.
HE	Number of compromises	BS detects one with probability 1, one node compromise. If both parent and child nodes are compromised, all the readings in the child's subtree can be falsified with probability 1.	$O(N)$	$O(1)$	$F = \begin{cases} 0 & c < 2 \\ N & c \geq 2 \end{cases}$
CPS	Optimally secure or not.	An adversary is unable to induce BS to accept any aggregation result which is not already achievable by direct data injection.	$O(N \log^2 N)$	$O(\log^2 N)$	$F = 1$

**Table 1- Comparison of aggregation algorithms.**

## REFERENCES

- Cárdenas A.A., Baras J.S. and Seamon K., A Framework for the Evaluation of Intrusion Detection Systems. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, Oakland, California, May 2006, 63-77.
- Chan, H and Perrig, A and Song, D, 2006: Secure hierarchical in-network aggregation in sensor networks. In *Proceedings of the 13<sup>th</sup> ACM Conference on Computer and Communications Security*.
- Du, W, Deng, J, Han, Y and Varshney, PK, 2003: A witness-based approach for data fusion assurance in wireless sensor networks. In *Proceedings of the IEEE Global Telecommunications Conference*, 1435-1439.
- Gaffney J. E. and Ulvila J. W., Evaluation of Intrusion Detectors: A Decision Theory Approach. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, Oakland, California, 50-61.
- Hu, L and Evans, D, 2003: Secure aggregation for wireless networks. In *Workshop on Security and Assurance in Ad Hoc Networks*, 384-392.
- Mahimkar, A and Rappaport, T, 2004: SecureDAV: A secure data aggregation and verification protocol for sensor networks. In *Proceedings of the IEEE Global Telecommunications Conference*, 2175-2179.
- Przydatek, B, Song, D and Perrig, A, 2003: SIA: Secure information aggregation in sensor networks. In *Proceedings of the 1<sup>st</sup> International Conference on Embedded Networked Sensor Systems*, 255-265.
- Wagner, D, 2004: Resilient aggregation in sensor networks. In *Proceedings of the 2<sup>nd</sup> ACM Workshop on Security of Ad Hoc and Sensor Networks*, 78-87.
- Yang, Y, Wang, X, Zhu, S and Cao, G, 2006: SDAP: A secure hop-by-hop data aggregation protocol for sensor networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 356-367.