REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE March 29 2002	3. REPORT TYPE AND DATES COVERED Interim Progress Report; May 01 to Dec 31, 01

4. TITLE AND SUBTITLE Distributed Immune System for Wireless Networks Information Assurance	5. FUNDING NUMBERS G DAAD19-01-1-0494
6. AUTHOR(S) J. Baras (PI), C. Berenstein (I), A. Ephremides (I), V. Gligor (I), R. Liu (I), H. Papadopoulos (I), N. Roussopoulos (I)	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Maryland, Office of Research Admin, 2100 Lee Building, College Park, MD 20742	8. PERFORMING ORGANIZATION REPORT NUMBER 1
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U. S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSORING / MONITORING AGENCY REPORT NUMBER 42340-CI-CIP

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The research program focused on developing innovative distributed methods and algorithms that take advantage of the special nature of wireless networks to improve assurance and security, while keeping disadvantages of wireless to minimum. The goal of the research is to design ‘robust’ information assurance systems. There are three interrelated thrusts: (1) Distributed Autonomous Immune Systems; (2) Assurance Via Distributed Physical Layer Signal Processing and Routing; (3) Distributed Computing Formalisms and Systems. Technical accomplishments for the reporting period: Developed formal models for intrusions based on attack trees (Extended Finite State Machines) and algorithms for estimating their parameters from network monitoring data; Developed algorithms for intrusion detection using Support Vector Machines; Initiated use of "tomographic" tools in a graph to find if a network has been “broken” into (monitoring data can be understood as samples of 2-D or 3-D tomographic data in hyperbolic space); Developed pseudochaotic signaling for robust and private communications over multiple access channels; Developed concept of layer integration for security, by making upper-layer protocol choices (MAC and routing) contingent on QoS at physical layer; Developed initial distributed trust establishment algorithms for MANETs using “swarm intelligence”; Developed a scalable media-dependent key distribution scheme using multimedia data embedding for group communications.

14. SUBJECT TERMS Distributed immune systems, wireless networks, attacks, intrusions, intrusion detection, layer integration, key management, multicast security, support vector machines, Internet tomography, pseudochaotic signaling for security, distributed trust, aggregates computation, shadow security network	15. NUMBER OF PAGES 22 16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UL**

Report Documentation Page (SF298) (Continuation Sheets)

(1) List of papers submitted or published under ARO sponsorship during this reporting period

(a) Manuscripts submitted, but not published

Y. Sismanis, A. Deligiannakis, N. Roussopoulos and Y. Kotidis, "Dwarf: Shrinking the PetaCube", to be published *Proc. ACM SIGMOD International Conference on Management of Data 2002*, Madison,Wisconsin, June 2002.

Laurent Eschenauer, Virgil Gligor and John S. Baras, "On Trust Establishment in Mobile Ad-Hoc Networks", presented in the 10th International Workshop on Security Protocols, April 2002, Cambridge, UK; to be published in *Security Protocols*, Lecture Notes in Computer Science, Springer, 2002.

John S. Baras and Maben Rabi, "Intrusion Detection with Support Vector Machines and Generative Models", submitted to *Information Security Conference 2002*, Sao Paolo, Brazil, September 30-October 2, 2002.

W. Trappe, J. Song, R. Poovendran, and K. J. R. Liu, "Key Management and Distribution for Secure Multimedia Multicast," to appear in *IEEE Transactions on Multimedia*.

Y. Sun, W. Trappe, and K. J. R. Liu, "A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks," submitted to *IEEE/ACM Transactions on Networking*.

M. Wu and Y. Mao, "Communication-Friendly Encryption of Multimedia", submitted to *IEEE Multimedia Signal Processing Workshop* (MMSP'02), May 2002.

H. Papadopoulos and Yeong-Sun Hwang, "Pseudochaotic Spread Spectrum Systems for Robust and Private Communication over Wireless Multiuser Channels", in final preparation to be submitted.

(b) Papers published in peer-reviewed journals

None

(c) Papers published in non-peer-reviewed journals or in conference proceedings

W. Trappe, J. Song, R. Poovendran, and K. J. R. Liu, "Key Distribution for Secure Multimedia Multicasts via Data Embedding," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 2001.

Y. Sun, W. Trappe, and K. J. R. Liu, "An Efficient Key Management Scheme for Secure Wireless Multicast," in *Proc. 2002 IEEE Int. Conference on Communications*, April 2002, New York City.

(d) Papers presented at meetings, but not published in conference proceedings

None

(2) Scientific Personnel supported by this project and honors/awards/degrees received

Dr. John S. Baras (Principal Investigator)
Dr. Carlos A. Berenstein (Investigator)
Dr. Anthony Ephremides (Investigator)
Dr. Virgil Gligor (Investigator)
Dr. K.J. Ray Liu (Investigator)
Dr. Haralabos Papadopoulos (Investigator)
Dr. Nicholas Roussopoulos (Investigator)

Dr. Vahid Ramezani (Postdoctoral Fellow)

Shah-An-Yang (Faculty Research Assistant)

Irena Bojanic (Graduate Research Assistant)
Alvaro Cardenas (Graduate Research Assistant)
Nitin Chandrachoodran (Graduate Research Assistant)
Matt Davies (Graduate Research Assistant)
Farshad Faroozan (Graduate Research Assistant)
Yeong-Sun Hwang (Graduate Research Assistant)
Song Li (Graduate Research Assistant)
Yinian Mao (Graduate Research Assistant)
Behnam Neekzad (Graduate Research Assistant)
Maben Rabi (Graduate Research Assistant)
Svetlana Radosavac (Graduate Research Assistant)
Yan Sun (Graduate Research Assistant)
Dimitrios Tsoumakos (Graduate Research Assistant)
Sudhir Varma (Graduate Research Assistant)
Lige Yu (Graduate Research Assistant)

(3) Report of Inventions

J. Song, W. Trappe, R. Poovendran and K.J.R. Liu, "A Dynamic Key Distribution Scheme Using Data Embedding for Secure Multimedia Multicast," U.S. and international patent application filed June 2001, PCT/USo1/19715.

(4) Scientific Progress and Accomplishments

Our research focused on the overall theme of the program, which is the development of innovative distributed methods and algorithms that take advantage of the special nature of wireless networks to improve assurance and security, while keeping disadvantages of wireless to minimum. Our goal is to design 'robust' information assurance systems, i.e systems capable to maintain assurance even under high levels of noise and node capture or destruction. The research program is organized around three interrelated thrusts:

(1) Distributed Autonomous Immune Systems
(2) Assurance Via Distributed Physical Layer Signal Processing and Routing
(3) Distributed Computing Formalisms and Systems

Our research on methods, algorithms, modeling and analytical methods is supported by:

- Mobile wireless network simulation testbeds
- Real experimentation with mobile wireless network testbeds

Our research plan is to investigate the following problems in an integrated manner:

- Automated vulnerability assessment
- Automated compromised subnetwork containment
- Pro-active intrusion and anomalous behavior detection
- Automated classification of intrusions and anomalous behavior patterns
- Automated and distributed storage and distribution of intrusion and anomalous behavior patterns
- Autonomous deployment of passive/active methods for intrusion defense
- Autonomous deployment of schemes assuring continuous operation at acceptable assurance levels
- Trade-off analysis between detection performance and false positives *vs* complexity and speed of response
- Robustness and resilience of the proposed assurance schemes
- Integration of transmission and traffic flow security with key generation/management and authentication

This integration will be achieved by innovative ideas and schemes that focus on the following principles: Distributed automatic classification of intrusions in real-time; Automatic generation of responses for containing and nullifying an intrusion faster than it spreads; Attacking intrusions close to the 'network edge'; Utilization of synergy between physical layer and network layer assurance schemes; Hierarchical methods and schemes in both the physical and logical domain for efficiency and scalability. Furthermore we have adopted a "systems view" of security and information assurance; that is security and information assurance belongs to network management and control.

Major motivation for our methods and ideas comes from: the operational principles of biological immune systems; recent successful development of 'digital immune systems' for the protection of commercial networks from virus attacks; recent advances in complex waveform generation which can be profitably utilized to secure wireless communications in a variety of yet unexplored ways.

Thrust 1:
Distributed Autonomous Immune Systems

In this thrust we are investigating the following topics:

- *Fast innate and adaptive immune systems*
- *Group authentication and multiparty key protocols in dynamic groups*

Fast Innate and Adaptive Immune Systems

We investigated the key principles that must govern the development of systems for information assurance capable not only to provide authentication and security but also to detect and defend against information assurance attacks. The principal objective for this research is to develop a distributed intrusion and anomalous behavior detection system that can detect such abnormal network behavior early on, and as close as possible to the first node(s) that has(ve) been attacked or compromised. The system then will autonomously generate a response to either isolate temporarily the compromised components or to attack and nullify the effects of the intrusion attack and re-establish normal operations.

During the reporting period we initiated investigations of the architecture of such systems and developed analytical formulations of the key problems. The proposed 'immune system' will in general have two components and the key properties outlined below:

(i) An 'innate immune' component which must be capable of detecting in a distributed manner the presence of a high proportion of intrusions that are unknown to it specifically: The system should recognize previously unknown intrusions and attacks of all types, including file infectors, masquerading, denial of service, etc.; This innate component detects *broadly defined* components or parts of an intrusion attack or anomalous behavior.

(ii) An 'adaptive immune' component which should be capable of automatically deriving from a single sample of anomalous behavior a 'prescription' for detecting and, if possible, removing or nullifying all instances of the intrusion or anomalous behavior: The detections and classifications performed are of the type needed *to identify specific intrusions* or abnormal behaviors; The system must deliver this prescription to the afflicted node(s) and must also facilitate dissemination of the prescription to other nodes both in the neighborhood and in the entire network at large; The system must be capable of detecting an intrusion and deriving and disseminating prescriptions on a time scale faster than the intrusion or anomalous behavior can spread.

During the reporting period we formulated broadly defined intrusion detection problems as change detection problems with very limited data. We investigated and developed various formal models for intrusions and proved equivalences between different types of models when such equivalences exist. More details on this part of the research is provided in Thrust 3 below. We used *attack trees* as models for attacks and established their equivalence to Extended Finite State Machines (EFSM). For simpler attacks FSM models suffice. We also established the equivalence of these models to Hidden Markov Models (HMM). We developed algorithms for estimating the parameters of these models from network monitoring data; EM type algorithms and modifications.

We investigated the use of adaptive clustering methods in order to quickly classify new monitoring data into either known intrusion types or new and unknown intrusions. We investigated in particular extensions of deterministic annealing, learning vector quantization, pairwise clustering and classification based on support vector machines (SVM). Thus the formulation at hand now divides the design of such systems into four components: (a) the estimation of parameters and structure of intrusions modeled as attack trees, extended finite state machines or HMMs; (b) the efficient storage of such learned models in a distributed database; (c) detection and classification of intrusions for which models exist in the database from new network monitoring; (d) detection, classification of new (previously unknown) intrusions and compaction of the database of models to avoid accumulation of errors. We also initiated investigations for the development of algorithms to extract key strings of bits that characterize various intrusion and anomalous behavior patterns (digital signatures), in particular the simplest possible ones.

Intrusion Detection with Support Vector Machines and Generative Models

We investigated the development of algorithms for intrusion detection using Support Vector Machines (SVM). Based on the formal models we developed (see Thrust 3 description below) intrusion detection was formulated as the problem of detection and classification of symbolic strings (attack tree model), given small amounts of data. Our main objective in this research is to combine use of formal models (like attack trees) and parallel clustering algorithms of the Support Vector Machine (SVM) type, to develop fast distributed algorithms for identification of components of intrusions and abnormal behaviors. Our approach and methods are somewhat inspired from similar successful combination of formal (logic) models and SVM in genomics and proteomics, and in particular for the problem of partial identification using side logical information. These are the so-called "generative models based SVM" schemes. They have excellent learning properties for unknown patterns and generalization capabilities.

We addressed the problem of detecting intrusions in the form of malicious programs on a host computer system by inspecting the trace of system calls made by these programs. We used 'attack-tree' type generative models for such intrusions to select features that are used by a Support Vector Machine Classifier. Our approach combines the ability of an HMM generative model to handle variable-length strings, i.e. the traces, and the nonasymptotic nature of Support Vector Machines that permits them to work well with small training sets. The central problem is to decide whether a computer program that runs on a single host computer system is a *normal* program that does not compromise the security of the host or a *malicious* program that is a threat to the host. To decide on the nature of a program, we examine the string (trace) of system calls that it makes and operate under the assumption that this trace contains all the information we need to make this decision. To be able to learn the decision rule, we are presented with a set of labelled training examples $T = \{(Y_i, L_i)|\ i = 1, . . . , T\}$. The traces Y_i are elements of $\boldsymbol{Y}$, which is the set of strings formed from the set of all possible system calls $\boldsymbol{Y} = \{y_1, y_2, . . . , y_s\}$. L_i

are labels from the set *{*0*,* 1*}*, 0 being the label for a normal trace and 1 for a malicious trace. The problem then is to come up with a rule *R*, based on the training set, that attempts to minimize the probability of misclassification (*Pe*). What is of more interest to system administrators is the trade-off between the Probability of Detection P_D, and the probability of False Alarms P_F that the classifier provides. A relatively large number of training examples especially of programs that are malicious is hard to come by. Therefore we seek to use non-parametric discriminative classifiers such as Support Vector Machines that, of all learning techniques are designed to work well with small training sets. The raw observations are members of the set ***Y**,** on which we have no suitable, ready-made distance metric that could be used to derive a kernel function for the Support vector machine. This is where we appeal to generative models and use them to extract a real-valued feature vector that resembles the Fisher score.

The malicious nature of a program is due to the presence of a subsequence, not necessarily contiguous, in its trace of system calls. For the same type of attack on the host, there are several different combinations of system calls that can be used. Furthermore, innocuous system calls or sequences can be injected into various stages of program execution (various segments of the traces). Thus the intrinsic variety of attack sequences and the padding with harmless calls leads to a polymorphism of traces for the same plan of attack. Real attacks have a finite (and not too long) underlying *attack* sequence of system calls because they target specific vulnerabilities of the host. This and the padding are represented in a 'plan of attack' called the *Attack Tree*. The basic attack scheme encoded in the Attack Tree is not changed by modifications such as altering the padding scheme or the amount of padding(time spent in the padding nodes). Given an attack tree, it is straightforward to find the list of all traces that it can generate. But given a trace, we don't have a scheme to check if it could have been generated by an attack tree A without searching through the list of all possible traces generated by the set of attack trees (an exponentially large set) . Our intrusion detection scheme needs to execute the following steps: (1) Learn about A from the training set T ; (2) Form a rule to determine the likelihood of a given trace being generated by A . These objectives can be met by a probabilistic modeling of the Attack Tree. Given an Attack Tree A , we can set up an equivalent Hidden Markov Model H^1 that captures the uncertainties in padding and the polymorphism of attacks. We prescribe the statistics of state transitions (with time replacing the position index along a trace) to reflect the edge structure of A and to also reflect the duration of stay in the vulnerability and padding nodes. The only allowed state transitions are the ones already in A and self-loops at each of the states. The picture is completed by defining conditional output probabilities given the state of system calls. We can write down a similar HMM for the normal traces also: H^0.

We would like to represent the probabilities for the above HMMs as functions of some vector θ of real-valued parameters so as to be able to use the framework SVMs supported by generative models. We used these parametric HMMs to derive a real valued feature vector of fixed dimension for these variable length strings that enables us to use Support Vector Machines for classification. Since we are dealing with variable length strings, we would like to extract the features living in a subset of a Euclidean space on which kernel functions are readily available enabling use of Support Vector Machines. We used the

Fisher score as the feature vector. We have used variants of the Expectation Maximization (EM) algorithm to estimate the HMM parameters, and based on them the Fisher score. Even though we face difficulties such as a large number of unknowns, a relatively small training set, and the problem of settling on a local optimum point in the EM algorithm, we are banking on the robustness of the Support Vector Machine classifier that uses the parameters of the generative model.

The overall scheme executes the following steps:

1. For the given T_1 attack traces of system calls Y_i, we estimate using the EM algorithm a HMM model H^1 for an attack with n_1 states.
2. For given T_0 normal traces of system calls, Y_i, we estimate a HMM model H^0 for the normal situation with n^0 states.
3. We compute the Fisher scores for either a signature-based intrusion detection or a combined signature and anomaly based intrusion detection.
4. Using the Fisher scores we train a SVM using either one of the kernels (Gaussian, Polynomial, Fisher).
5. Given a test trace of system calls Y, we let the SVM classifier decide as to whether the decision should be 1 (attack) or 0 (normal). The Fisher scores of Y are computed and entered in the SVM classifier.

We performed numerical experiments on a subset of the data-set for host based intrusion detection from the University of New Mexico. In the experiments, we tried various kernels in the SVMs. The performance evaluation is based on the computation of several points of the receiver operating characteristic (ROC) curve of the overall classifier; i.e. the plot of the curve for the values of the probabilities of correct classification (detection) P_D *vs* the false alarm probability P_F. In addition to the above scheme using the Fisher scores, we implemented and tested an algorithm that uses a simpler feature that avoids the estimation of the gradient of the likelihoods; a vector of the number of occurrences of various *gappy bigrams* in a trace. In our experiments, we compared the performance results for various forms of these schemes and selected components of the schemes.

Group Authentication and Multiparty Key Protocols in Dynamic Groups

In contrast to multicast groups, dynamic peer groups (DPGs) tend to be smaller (involving a hundred or a few hundred members), flat (*vs* the hierarchical groups often encountered in multicast communications), and employ many-to-many communication patterns (*vs* the one-to-many communication patterns prevalent in multicast communications). We initiated investigation of key management and authentication for DPGs. We focused particularly on investigating these problems with the additional requirement of good performance under rapid changes in the network topology, group membership and severe bandwidth and energy constraints. We initiated the development of distributed algorithms that do not rely on the existence of centralized completely trusted third parties: the so-called key agreement type of key management. Such algorithms involve two phases: an initial key agreement phase and an auxiliary key agreement phase. We focused on algorithms that have low computational and time complexity and can operate efficiently when topology changes rapidly.

Initial improvements were achieved by employing a combination of methods: fractional keys, logical tree keys, key multicasting, key caching, adapted to the mobile wireless environment. The resulting algorithms promise to be of low complexity and amenable to distributed implementation. We also initiated investigations of information assurance for DGPs.

Thrust 2:
Assurance Via Distributed Physical Layer Signal Processing and Routing

In this thrust we are investigating the following topics:

- *Advanced signal processing for channel and communication assurance and authentication*
- *Wireless multimedia security, authentication, and dynamic key management*
- *Use of covert channels*
- *Simultaneous selection of access control and routing*
- *Securing mobile adhoc networks*

Pseudochaotic Spread Spectrum Systems for Robust and Private Communication over Wireless Multiuser Channels

We investigated novel approaches to secure communication over fading channels that exploit the sensitive dependence of chaotic systems to initial conditions in conjunction with the deleterious nature of the fading channel to achieve privacy. Our focus is on designing a novel class of pseudochaotic (PC) spread spectrum systems for private multiuser communication over wireless channels. These spread spectrum systems exploit linear modulation of a digital information-bearing signal on a pseudochaotic sequence, i.e. a sequence generated by iterating an initial condition through a pseudochaotic mapping. Privacy is achieved by providing side information to the intended users in the form of making available knowledge of the initial condition. These systems are attractive alternatives to conventional systems, as they can provide Pr(e) performance advantages to intended users by exploiting the inherent sensitivity to initial conditions of chaotic systems with minimal increase in transmitter and intended receiver complexity relative to conventional systems and affordable cost in key distributions.

Pseudochaotic spread spectrum systems possess a number of properties that make them appealing candidates for private multiuser communication over fading channels. In addition, these systems are considerably less computationally intensive than other privacy achieving methods. As a result, they are very attractive candidates for mobile wireless systems where all signal processing operations must be performed in real-time and with limited processing power. Pseudochaotic spread spectrum systems are also naturally suited for multiuser communication systems. Indeed, due to the properties of the underlying pseudochaotic sequences, the resulting transmitted sequences have broadband spectra and the resulting time-varying individual user signatures can be readily designed to have low cross-correlations by proper choice of a *single* pseudochaotic map and for almost all distinct choices of initial conditions. In addition, unlike their conventional

CDMA counterparts where the same signature sequence is used to modulate all the symbols communicated by a particular user, the inherent time-varying nature of the pseudochaotic signatures provides pseudochaotic spread spectrum systems with significantly higher privacy benefits.

The class of pseudochaotic systems we have been exploring involve one-dimensional (1D) iterative mappings, that are in some sense quantized versions of 1D chaotic maps. In this reporting period we focused on parametrized families of piece-wise linear maps which exhibit chaotic behavior. Examples of interest include the families of tent maps and Bernoulli maps. For properly selected parameter values in each of the families, the resulting 1D maps can exhibit very diverse behavior, including stable points, eventually periodic and quasiperiodic sequences, and chaos.

Our main objective is to develop systematic methods for assessing the performance characteristics and trade-offs between intended and unintended receivers for a large class of PC maps and a variety of channel models, as a function of processing gain, SNR, and number of users in the system. During this period our efforts have been mostly focused on designing tools for characterizing the differences in attainable performance between intended and unintended users in single-user settings (corresponding to only one transmitting user), as a function of processing gain and SNR for a large class of PC maps. In particular, we have focused on determining the performance characteristics of spread spectrum schemes with signatures generated by various families of PC piece-wise-linear maps, in the context of signaling over additive white Gaussian noise and frequency nonselective fading channels.

In general, the intended receiver may exploit its knowledge of the initial condition to reconstruct the PC sequence used at the transmitter. As a result, optimized detectors for various different channel models and detector complexity criteria, can, in many cases, be derived as extensions of their conventional CDMA counterparts. The detector design in the unintended receiver case is a composite detection problem - thereby distinctly different from the intended user setting-and requires distinctly different signal processing techniques. In general, due to the chaotic nature of the modulation waveform, the complexity of the optimal unintended receiver grows exponentially with the processing gain L. Although, this is an attractive feature of our schemes, it makes direct analysis via simulation of the optimal receiver impractical for even moderate values of L. We therefore have to rely on developing tight upper and lower bounds as a means of evaluating the performance characteristics of unintended receivers.

We have successfully applied these techniques in the analysis of the pseudochaotic spread spectrum systems with signatures generated by tent and Bernoulli maps, over frequency nonselective channels. The Bernoulli and tent sequences share the same energy profile. This fact can be exploited to show that these two maps share identical performance characteristics from the point of view of intended receivers. In addition these two maps possess identical sensitivity to initial conditions (same Lyapunov exponent). Detector design and performance characteristics for unintended users, are remarkably different in the case the signatures are generated from Bernoulli map from their tent map counterparts. Specifically, unlike the tent signature setting, symbol-by-symbol detectors are clearly suboptimal in the setting involving Bernoulli signature sequences, as any such detectors can be shown to provide no information about the transmitted symbol. Sequence detection is required to optimally exploit the available information in this setting. We have developed tight upper and lower bounds on the Pr(*e*) of the optimal (unintended) sequence detector for spread spectrum systems with tent and Bernoulli signatures.

A special class of piece-wise linear maps that we are currently exploring in this project is that of Markov maps. Both the tent map and the Bernoulli map are special cases of Markov maps. Markov maps are amenable to characterizations that are typically a lot harder for other maps. As part of our preliminary work, we have exploited these techniques to construct mappings that lead to spread spectrum systems with superior performance characteristics to their Bernoulli map counterparts. These mappings have higher Lyapunov exponents than Bernoulli sequences. Another viable approach we are currently exploring to systematically construct and explore families of piece-wise linear mappings is based on the notion of topological equivalence and pertains to applying an invertible transformation on other piece-wise linear maps such as on tent or Bernoulli maps. We are currently focusing on piece-wise linear invertible transformations with endpoints that include those of the piece-wise linear map that is transformed. Such transformations provide an attractive framework for designing maps with objective to select the map from a family that yields the worst-case unintended-receiver Pr(*e*) characteristics.

Key Distribution for Secure Multimedia via Data Embedding

In order to secure multicast communications, all the members of the group share a common session encryption key (SK). When changes in the group membership occur, the SK must be changed. In order to update the SK, a party responsible for distributing the keys, called the group center (GC), must securely communicate the new key material to the valid users. This task is achieved by transmitting rekeying messages that use key encrypting keys (KEKs) to encrypt and distribute new keys. There are two types of channels available for distributing the rekeying messages. The first approach is to use a *media-independent channel*. By this we mean that a separate channel needs to be used to convey the keying material. We propose employing a *media-dependent* approach to transmitting the rekeying information that is accomplished using steganographic methods. The primary advantage of using data embedding to convey rekeying messages compared to a media independent channel is that data embedding provides an additional layer of security that hides the presence of rekeying messages from potential adversaries. In the conventional approach of using a media-independent channel to convey the rekeying messages, an adversary can observe the external channel and determine information about the membership dynamics of the multicast service, such as the rate at which members join and leave the service as well as being able to infer information about the group membership. From a security point of view, this provides valuable information to a potential adversary. In comparison, data embedding provides *covert* information transferal, whereby the bit rate of the multimedia source is maintained and it is impossible for an eavesdropper to measure information regarding the occurrence of a rekeying operation.

We have developed a new key message format, using a residue based format and logical tree keys, that can be used to update and maintain keys used for securing multimedia multicast. Consider a group of n multimedia users who will share a multimedia multicast. In a simple key distribution scheme for n users, each user has two key encrypting keys K_j and K_ε, and the session key K_s. The session key K_s is used to encrypt bulk quantities of multimedia content. The KEK K_ε is the root KEK and is used to encrypt messages that update K_s. The remaining keys are KEKs that are used to protect updates of $K_{\varepsilon'}$. Each key has length B bits. We use parametric one-way functions (POWF). When the multicast group is very large, the residue based message format has severe computational and communication overhead associated with member departures because the size of the rekeying message increases linearly with the number of users n. One popular approach to improving scalability is to distribute the keys according to a tree structure. A binary tree is usually employed, though in the general case the tree can be an a-degree tree. Attached to the tree above the root node is the session key K_s. Each node in the tree is assigned a KEK which is indexed by the path leading to itself. The symbol ε is used to denote the root node. Each user is assigned to a leaf and is given the KEKs of the nodes from the leaf to the root node in addition to the session key. The residue-based rekeying format can be used with a key tree for improved resource usage. When a member leaves the group, multiple keys become invalidated because that user shares these keys with other users. The most efficient approach to updating the keys is to update them from the leaf node to the root node (bottom-up). The total amount of communication needed to update the entire system of keys when using an a-ary tree is $(B + 1)(a \log_a n - 1)$.

The $O(\log n)$ communication of most tree-based schemes makes the use of a tree-based scheme essential when the group size is several thousand or more users. Suppose we transmit the rekeying information for this scheme via a media-dependent channel. If we denote R as the embeddable channel rate (in bits/second), B_{KEK} to be the key length of a KEK, B_s to be the key length of the session key, B_m the bit length of the random seed, and B_{emb} to be the key length governing the data embedding rule, then the amount of time needed to update the entire system of keys is $T = (C_{bu}B_{KEK} + B_s + B_{emb} + B_m)=R$. Since T is related to the bit size of each of the keys, it is therefore related to the security levels protecting the service. This amount of time corresponds to the amount of time the departing member may still enjoy the service before no longer being able to decode the video stream. If we desire to increase the level of protection of the multimedia, then B_s must be increased, which leads to an increase in the amount of time needed to refresh the entire set of keys.

We used data embedding in conjunction with tree-based rekeying employing the new message format in various simulation experiments. No visible distortion was observed, and we calculated the amount of time needed to refresh the entire network of keys for a binary tree with $n = 2^{20}$ or roughly one million users. We took $B_{KEK} = 56$ bits, $B_s = 56$ bits, $B_m= 56$ and $B_{emb} = 20$ bits as the bit lengths for the various keys. These values for

B_{KEK}, B_s and B_m were chosen since they correspond to the key size of the popular block cipher *DES*. The resulting times needed to refresh the keys show an inverse relationship with the amount of bits embedded per frame. Using these relationships (and curves), one can determine the necessary embedding rate needed to refresh the keys in time *T*. For example, if we have a video service of QCIF images with a frame rate of 20 frames/second, and desire to refresh the keys during member departure in $T = 5$ seconds, then 25 bits must be embedded per frame.

Key Management Scheme for Secure Wireless Multicast

In tree-based multicast key management schemes, most rekeying messages are only useful to a subset of users, who are always neighbors on the key management tree. We designed a key management tree that matches the network topology in such a way that the neighbors on the key tree are also physical neighbors on the network. Additionally, by delivering the rekeying messages only to the users who need them, we may take advantage of the fact that the key tree matches the network topology, and localize the delivery of rekeying messages to small regions of the network. This lessens the amount of traffic crossing portions of the network that do not have users who need to be rekeyed. In order to accomplish this, it is necessary to have Time to refresh all keys (seconds) the assistance of entities that would control the rekeying message transmission, such as the Base Stations (BS) in a Cellular Wireless Network. In this context, a BS has the knowledge of whether the rekeying messages are useful for its users, and the rekeying messages are first sent to BS's, which would decide whether to forward the message to the users under them. Based on the discussion above, we designed a key management tree that matches that network topology in two steps: First, design a subtree for the users under each BS. Those subtrees are called *user-subtrees*. Next, design a subtree which governs the key hierarchy between the BS's and the SH and shall be called *BS-subtree*. Since the combined key management tree depends on the network structure, we called it a Topology-Matching Key Management (TMKM) tree.

Traditional key management trees are independent of the network structure, and we call them Topology Independent Key Management (TIKM) trees. When using a TIKM tree, rekeying messages are sent to every user, i.e. broadcast by all BS's. When using a TMKM tree, rekeying messages are broadcast by only a subset of BS's. Let S_1 denote the number of the messages multicast to the BS's, and S_2 denote the number of the messages broadcast by the BS's. The measurement of communication burden, wire-line cost C_{wire}, wireless cost $C_{wireless}$, and total cost CT , are defined as $C_{wire} = E[S_1]$, $C_{wireless} = E[S_2]$, and $C_T = \gamma C_{wireless} + (1-\gamma)\, C_{wire}$, where γ $(0 < \gamma < 1)$ is the wireless weight, which indicates the importance of considering the wireless cost. Given the wireless weight, both the TMKM and TIKM trees should be designed to minimize the total communication cost, C_T . We have proposed the design of user subtrees and BS subtrees by introducing a new tree structure, the **(*a; L;* x)-logic tree**, which allows the key management tree to match the network topology. The (*a; L;* x)-logic tree has L +1 levels. The upper *L* levels, which form a symmetric subtree with degree *a*, are fixed during the multicast service. The $(L + 1)$*th* level changes when users join/leave. Users are attached to the upper nodes randomly. By allowing the lowest level of the key tree to have varying degree, the ALX tree maintains its structure as users join and leave and can be optimized based on the statistics of the number of users. The performance of the ALX tree is very close to the performance lower bound of any fixed degree tree; corroborated through simulations.

In mobile environments, the user will subscribe to a multicast service under an initial host agent, and through the course of his service undergo *handoff* to different base stations. When a user travels from one cell to another cell, this user may need to be moved on the key tree, which requires extra rekeying messages to be sent. To lessen the undesired effect of handoff, we designed an efficient handoff scheme that is suitable for topology-matching key management tree.

We simulated a simple cellular wireless network with 12 concatenated cells for different users' joining/leaving behavior. We define the *performance ratio* η as the total communication cost of the TMKM tree divided by the total communication cost of the TIKM tree. When η is less than 1, the TMKM tree has smaller communication cost than the TIKM tree. In addition, smaller η indicates a greater advantage of the TMKM tree over the TIKM tree. Our simulation and analysis results for the performance ratio η, for large variations in parameters such as the maximum speed of mobile users, the radius of the cellular cells, the user join rate, and the average time duration when a user is staying in the multicast service, demonstrate that the communication cost of the TMKM trees is only 33%-45% of the communication cost of the traditional key management trees that are independent of the network topology.

Covert Wireless Channels

We have developed a unique view of the ubiquitous nature of wireless transmission that enables us to use them for covert transmission in a natural way. We have focused our attention on sensor networks in which data can be fused (based on the assessment of a "risk" metric) and then covertly transmitted through the use of timing, power level, or choice of routes. In fact, the very decision to fuse data or not may represent an additional form of covert transmission. In our past work we analyzed a unique wireless covert channel that employs artificial packet collisions during the execution of conflict resolution algorithms as a means of transmitting information covertly. The basic idea was to exploit the ubiquitous nature of wireless transmission (through its broadcast and multiple access properties).

In the work reported here, we rely on the same fundamental premise but in a totally different way. We use the case of a sensor network as an example but our ideas apply to other ad-hoc wireless networks as well. We allow every node the choice of either transmitting all the data it has accumulated (via measurement or via reception from neighboring nodes) to a next neighbor according to a given routing algorithm or fusing the data it has received (and/or measured) and sending on a compressed version of them. In this manner, each node may transmit covert information that is coded into its decision to fuse or not. In fact, we permit multiple levels of fusion/compression and, thus, enable the covert transmission to take place at a higher rate. In addition, the actual transmission in terms of one of the assumed choices may be further utilized for further covert transmission by controlling either the timing or the power of the transmission as well as the choice of route to the eventual destination. In fact, as these choices affect overall network performance (as investigated in our second problem), it is possible to eventually integrate our work on this topic with the work on the next topic "exploiting vertical protocol integration for enhanced security" into a single and unified methodology for active and passive security.

Exploiting Vertical Protocol Integration for Enhanced Security.

By making upper-layer protocol choices (MAC and routing) contingent on QoS at physical layer can increase network robustness against threats such as jamming, denial of services, etc. We have first established an iterative algorithm for joint MAC/routing that allows network resilience (and, hence, increased robustness) in the presence of threats. We plan to use this joint control of MAC and routing as a vehicle for secure key distribution, via covert transmission. In our past work, we have argued that the inherent interdependencies among protocol layers dictates the joint design across multiple layers. We have focused on the lower three layers in which these interactions are strongest.

In the work reported here we take the blueprint for joint MAC/routing design that permits an iterative interplay between the allocation of bandwidth and the choice of routes and subject it to various threats that attack either the physical channel or the media-access control or the routing algorithm. We then studied how the joint choice of parameters across these layers increases the network's resilience (in an escape-valve manner), since the attack on one aspect of network operation enables the recovery of the network by means of adjustments to other aspects. In other words, if the MAC protocol is stressed, routing adjustments provide the network with an escape route. Similarly, if the routing protocol is under attack, MAC adjustments provide alternative escape routes.

Thrust 3:
Distributed Computing Formalisms and Systems

In this thrust we are investigating the following topics:

- *Formal methods for intrusion models*
- *Dynamic topology discovery and network tomography*
- *Distributed trust models for mobile wireless networks*
- *Active distributed databases, caching and prefetching of intrusion pattern models*
- *Cooperative intrusion detection databases with aggregates on a shadow security network*

Formal Methods for Intrusion Models

We investigated formal models for categorizing/analyzing intrusions. Developing formal methods for the representation, testing and storage of intrusion patterns and anomalous behavior patterns is both promising and critical. Such research for wireless mobile networks has not been undertaken previously. Within the telecommunications community particular attention has been given to three standardized formal description techniques: LOTOS, Estelle and SDL. The primary application has been protocol specification and validation. Recently communication services were modeled using formal methods and models used with field data to validate service behavior and quality. Formal models offer several advantages for problems of intrusion detection and information assurance. Our approach has centered on models employing *Attack Trees*. Attack trees are a promising and systematic method to characterize system security based on their ability to characterize a very large variety of network attacks.

We refine information about attacks by identifying the compromise of network security or survivability as the root of the tree. The ways that an attacker can cause this compromise iteratively and incrementally are represented as lower level nodes of the tree. A network typically has a set, or forest, of attack trees that are relevant to its operation. The root of each tree in a forest represents an event that could significantly harm the network's mission. Each attack tree enumerates and elaborates the ways that an attacker could cause the event to occur. Each path through an attack tree represents a unique attack on the network. The structure and semantics of attack trees are as follows. We decompose a node of an attack tree either as: a set of attack sub-goals, all of which must be achieved for the attack to succeed, that are represented as an AND-decomposition; or a set of attack sub-goals, any one of which must be achieved for the attack to succeed, that are represented as an OR-decomposition. Attack trees can be represented graphically or textually. A general attack tree is a combination of AND and OR nodes. Textual representation is simpler and clearer for complex attack trees. One generates intrusions from an attack tree by traversing the tree depth first. Thus an intrusion is represented through an attack tree as a symbolic string. Thus recognizing intrusions becomes a problem in recognizing "abnormal" symbolic strings

During this reporting period we developed formal models for intrusions based on attack trees and demonstrated their equivalence to Extended Finite State Machines and Hidden Markov Models. We also developed algorithms for estimating/classifying structure and parameters of attack trees from network monitoring data. We have initiated investigations on minimizing the resulting model complexities and in progressive updating of existing intrusion models. We also started exploring the relations between the complexity of these models, learning of these models, and performance of intrusion detectors/classifiers as described by the probabilities of correct classification (or detection) and miss-classification (or false alarms). Finally we investigated the consequences of polymorphism (a key characteristic of attacks) on these formal modeling techniques.

DynamicTopology Discovery and Network Tomography

The key question we have addressed is: how to produce a reliable and constant monitoring of the quality, reliability and, possibly, composition of ad-hoc networks, like that of troops communicating in the field, Internet groups, or pay-per-use Internet services. In particular, to find “cheap and fast” procedures for the early detection of attacks on the integrity of the network. The approach taken here is that, at least formally, the problem belongs to the fast evolving field of mathematical tomography. Moreover, it appears that the “correct” tomography is that taking place not in the Euclidean plane or space but almost certainly in a 2-D or 3-D real hyperbolic space.

There are several reasons why this seemingly far-fetched model seems to be the correct one. First, empirical evidence, suggests that the number of nodes of the networks of interest to this project have only a power law growth with respect to any point of the network. Second, recent and well publicized visualization work shows how these types of networks can be visualized in real hyperbolic space. In fact, there is a growing awareness of this fact in the scientific community.

What we have been investigating during this reporting period is how to go from the heuristic evidence to a more quantitative, mathematically correct, formulation of this relation. The corollary of this work will be that mathematical tools which were already developed, or are been developed, for related problems and other applications of tomography would then become available to study the question about networks raised above. Although the general principles are similar, it is clear that this way of studying networks should adapt to the first obvious distinction among networks: size. A small party of soldiers in the field or police officers in a manhunt, together with the base station, will form a small network, whose integrity can be checked by a number of ad hoc methods to study small graphs. The other extreme is the Internet or a portion thereof. In this case the number of nodes is both increasingly large and also variable. Thus the exact number of nodes is not a useful parameter. Instead, only partial information can be obtained by setting up monitors at a relatively small subset of the nodes. From the monitors, data can be collected and examined. The problem of discovering the detailed inner structure of the network from the collection of end to end measurements can be seen as a type of inverse problem, analogous to those arising in tomography, but with a strong discrete flavor. For instance, the problem of interest includes checking connectivity, finding largest components, tracking data traffic, assessing the performance of software/hardware and dealing with a variety of security issues. One of the ways to try to understand what's going on, is to visualize this large directed graph by laying it out in 3D Hyperbolic space or even 2D Hyperbolic space, since in these spaces the volume of a ball increases exponentially with the radius, as opposed to the familiar geometric increase of the volume of a ball in Euclidean 3-space, respectively 2-D Euclidean space. In fact, it was W.P. Thurston who first proposed to apply this model to study large graphs (1980).

In view of the several ways we can think about the graph associated to a network, the question is how to exploit these models to test the integrity of the network “from the outside”. If we think in terms of graph theory, we are led to consider a “chip firing” scheme, as studied by Fan Chung and collaborators. That is, we send “probes” (short spiky signals) from all the points accessible from the “outside” and the responses we get back generate an input-output map. The test “reduces” to study this map. The analogy with a well-known problem in material sciences is striking. It is usually called Electrical Impedance Tomography. In it one tries to determine whether a plate (or an airplane wing, or another metal component) has developed a crack which is not visible by simple inspection. The idea is to test the conductivity of the plate by sending electrical pulses at different boundary points of the plate and measuring how much of this disturbance remains at the other accessible boundary points of the plate. This is a sort of Neumann boundary value for the Laplace operator of the plate. The key point is how to go back and forth between our network problems (discrete) and the suggested paradigm. There has been interesting work in this type of questions in the recent past, but the detail "dictionary" that we are working on, has not been investigated much. The discrete questions can be found under the headings of *spectral graph theory* and *graph embedding*. On the direction of going from the continuous into the discrete, it falls into one could call discretization of Riemannian manifolds.

Another question that we are investigating is: how much "monitoring" do we need to do if we suspect the breakdown or compromise of the network has occurred in a "subregion" of the graph. Here we are guided by our previous work of on the localization of the Euclidean Radon transform. From this result one understands that the construction of well adapted multiresolution analysis will allow a significant saving in the amount of monitoring necessary to pinpoint the disruption, given that it occurred in a certain subregion known in advance by other means. Let us describe what localization means in the more familiar context of medical CT tomography. As it is well known, to image a planar cross section of the body one needs (theoretically) to use x-rays crossing through every point of the section in all possible directions, and one cannot reconstruct a smaller region by using only rays through that region. In our earlier work we showed that, up to a choice of baseline, it is in fact possible to reconstruct a smaller subregion, with data obtained over a slightly larger one, the key element was the use of wavelets. In particular, the complexity of the processing is proportional to the area of the region of interest, something crucially important in our application, where we expect quite often to be interested in monitoring in detail only a subregion of the total graph. As explained above, the determination of what is going on a graph can be attempted by the probabilistic method of firing chips. Basically, this is the discrete analogue of determining the conductivity of a circular plate (in 2-D, or solid volume in 3-D) by means of boundary measurements. In our earlier work we showed that the natural setting of this problem is in the hyperbolic plane, and that one can explicitly linearize the problem around the constant conductivity case in terms of the corresponding tomographic principle. For practical applications of the kind we have here we would like to prove the same type of localization theorem in the hyperbolic geometry as we did for the Euclidean Radon transform. One of the first things to do is to define a suitable Multi Resolution Analysis (MRA) on the hyperbolic unit disk such that arbitrary functions can be approximated using (hyperbolic) wavelets.

Distributed Trust Models for Mobile Wireless Networks

During the reporting period we investigated the notion of "trust" among entities engaged in various protocols as a set of relations established on the basis of a body of supporting (trust) evidence. In traditional networks, most trust evidence is established off-line, via potentially lengthy processes, and all of it is assumed to be valid on long terms and certain at the time when trust relations relying on it are exercised. The trust relations among entities are often inherited from those of the entities' “home” domains and follow hierarchies, such as naming hierarchies.

In contrast, few of these characteristics are prevalent for trust relations and trust evidence in mobile ad-hoc networks (MANET) beyond the fixed supporting “cores” of such networks. In particular, trust relations may have to be established using only on-line available evidence, may be short-term, largely peer-to-peer, where the peers may not necessarily have a relevant home domain that can be placed into a recognizable trust hierarchy, and may be uncertain. We have developed new formulations for such networks allowing for a substantial body of trust evidence to be: (1) generated stored and protected at various network nodes; (2) routed dynamically and autonomously where most needed; and (3) evaluated “on the fly” to substantiate dynamically formed trust relations. In particular, our formulations make possible that the management of trust evidence should allow alternate paths of trust relations to be formed and discovered using limited backtracking though the ad-hoc network, and should balance between the reinforcement of evidence that leads to “high-certainty” trust paths and the ability to discover alternate paths. We derived several design parameters for the management of trust evidence in mobile, ad-hoc networks by analyzing salient characteristics of peer-to-peer, file sharing protocols such as Freenet. We also obtained results of simulation experiments comparing different trust-evidence routing schemes. During this period we also initiated the investigation and exploration of “swarm-intelligence” based systems as a possible basis for trust-evidence routing.

Cooperative Intrusion Detection Databases with Aggregates on a Shadow Security Network

Intrusions follow well-defined patterns that can be detected using pattern-matching techniques on audit-trail information stored in the system. For example, series of attempts to login to a system or to “setid” inside a program can be detected by examining log messages resulting from system calls. An important question is then where in the network are these models stored and how are they moved around the nodes in response to the evolving intrusion or information assurance attack. Anomalous intrusions are detected by observing significant deviations from normal behavior. In large heterogeneous wireless networks there is no centralized audit-trail system that logs access and usage behavior that can be accessed in real-time to do the detection. The logs are distributed, heterogeneous, and the intermittent connectivity prohibits real time access for detection.

For the task in hand we have proposed a “shadow security network” (SSN) that is a substrate of the wireless networks for real-time delivery of log patterns, statistics, and control messages to the various interconnected nodes responsible for the security of a sub-network. SSN is an independent orthogonal network channel that could either run on its own platform or can be multiplexed with the normal data channels. Its whole purpose is intrusion detection and prescription (vaccine) for defending an attack. SSN is "broadcast" network delivering security data. A Cooperative Intrusion Detection Database (CID2) will consist of patterns observed, statistics aggregated by individual nodes, statistics observed across multiple nodes, and prescription to remedy attacks. CID2 will completely reside on the SSN and its pieces will continuously be distributed as a "broadcast" program.

Each participating node of the SSN consists of a Monitor, an Aggregator, a Broadcaster, and a Cache. The Monitor observes statistics on the SSN channel and decides which of these consider an intrusion. It uses the Aggregator to compute new aggregates. More specifically, the Aggregator combines the broadcast statistics received with local statistics from the node logs, updates the local cache, and passes them to the Broadcaster. The Broadcaster schedules them into its broadcast program along with possible prescriptions for the SSN. The Monitor is capable of executing prescriptions for either defending an attack or for invoking the Aggregator to start a new aggregation of statistics with a different accumulation point or a totally new aggregate view of the network.

In our work so far we have focused on the types of data to be collected and on the structures used to store the collected data. The data needed to support network intrusion detection and prevention falls in the following categories:

- Attempted Intrusions / Attacks
- Behavior statistics
- Statistics on the communication channel

Data on attempted intrusions/attacks stores information generated by the intrusion detection system that the current node of the network runs. Each such system typically produces warnings in a textual format of all the intrusive behavior it has observed. These statistics are then gathered, aggregated in multiple granularities and stored in our system. As a representative intrusion detection system we have selected SNORT from which we have created a filter that process its output and creates data files in the format needed by our system. Our system can be very easily used with ***any*** other system, providing that we know the format of the data it outputs, in order to create the appropriate filter. For the case of SNORT, the types of data we collect are:

Dimensions:

- Time: sec → min → hour → day
- Triggered Rule: Rule ID → Attack Type
- Source: IP/Node ID → Geographical Area
- Source Port
- Recipient: IP/Node ID → Geographical Area
- Target Port

Measure:

- Count(triggered rules)

Examples of queries that we can answer from this data are:

- "How many rules where triggered with target the local host, between 8:00 and 8:30 am and originated from the GeographicalArea (NW)?"

- "Which is the severity of the attacks suffered in the last 5 minutes?" Such a query can be easily answered providing we have a mapping from each identified attack to a severity metric.

After being able to answer such queries, it is very easy to provide rules based on the collected aggregates. These rules observe the value of specific aggregates and trigger specific actions (which in our case are the generation of "watch dog aggregates").

Behavior statistics aim at detecting a possible successful intrusion by observing significant deviations in the system usage over time. The data that we collect are:

Dimensions:

- Process Id/Process Name
- User Id → Group Id
- Start Time: sec → min → hour
- End Time: sec → min → hour

Measures:

- CPU usage
- Number of spawned processes
- Files opened/modified
- Memory Used (stack/total/shared)
- Network Connections

By running a data-mining algorithm, one can look for suspicious shifts in behavior and alert neighboring nodes about this. In this case, the node will also have to send the collected data that validate this suspicion. In this way the other nodes can initialize "watch dog aggregates" about the suspicious node.

Statistics on the communication channel have a two-fold benefit:

- They aim at observing abusive behavior of nodes in the communication channel
- They can be used later to fine tune the system even more

Abusive behavior of nodes is detected by observing two kinds of data: Conflicts in data transmission; Amounts of time a node was using the broadcast channel, along with the number of interesting items received from it.

In our work we decided to use as our basis the Cubetree Storage Organization (CSO) as the basic data structure. We had earlier shown that Cubetrees exhibit at least 10 times faster query performance when compared to indexed relations, half the storage a commercial relational system requires and at least 100 times faster update performance. Since no system has been shown to outperform the Cubetrees so far as far as update and query performance are concerned, we concluded that they constituted a good basis for our system. The performance characteristics obtained by CSO satisfy the first three characteristics that we deemed essential for our data structure to possess. Unfortunately, CSO was not programmed to consider the case of hierarchical dimensions. Thus, it could not be used directly to satisfy the needs of our system. We thus decided to augment the functionality of CSO and incorporated the use of hierarchical dimensions. We first created a metadata manager, which handles the mapping of a value from the base levels of a dimension to any other hierarchy level of the dimension. Then we modified the construction and the update processes of CSO to fit our needs. During the construction (update) phase, views containing at least one dimension mapped to a hierarchy level other than their base level are produced by the smallest (in size) *candidate* view that can produce them. When the smallest candidate view has been identified, then we use the metadata manager to map the values of the *rollup* dimension to one level higher. We have also explored new ways to minimize the storage and time required to materialize any views by exploiting the structural redundancies among the views.

In a wireless environment intrusion detection is harder because of the limited bandwidth for collecting and correlating the data in a single place. It is imperative that all nodes in a wireless network have a fast and reliable communication channel for data gathering information and prescription dispatching. We use Aircache, a hybrid data dissemination protocol from our earlier work, that is shown to be scalable and behaves extremely well under very severe situations of "panic" where the skew of the demand was directed towards the updates of the data. The aircache update channel is hierarchically organized so that the most important (or latest) have higher priority than those of less important (or stale) and channel division is configurable to the application's requirements. Many of the techniques employed in aircache are utilized in the SSN, but some have to be adapted to the specific requirements of wireless ad-hoc networks. Data aggregates in a compressed form are broadcast in the SSN while queries and specific node to node communication may be handled through standard point-to-point connections between nodes.

During the reporting period we have addressed the following issues:

- There is no distinction between clients and servers. All or some of the nodes hold data (aggregates, prescriptions, etc) and all (or most of them) use/update them from time to time. Moreover, the nodes perform special operations on broadcast data (add their data, change monitor granularity, etc). In our case then, there must be a decision on the broadcast items in a decentralized architecture.
- Each node has its own broadcaster that decides on what to broadcast, with what frequency and, most important, what order. What is the amount of data each node is allowed to put on its schedule? How often can it broadcast its data so that others also get their share?
- Furthermore, in our network there is a strong presence of the notion "severity". An issued prescription to deal with a possible intrusion is obviously highly ranked and should be broadcast more urgently than other data. Or, aging aggregates play a less important role than fresher ones. Watchdog aggregates are also "priority mail" for our system. How are these relative "severity functions" for each data item computed and executed on our channel? How is an agreement on behalf of the nodes reached? And if the decision taken is local, how is proper behavior ensured?

(5) Technology Transfer

We have held technical meetings and made technical presentations on these topics to several companies and Government Laboratories.