TAM: A Tiered Authentication of Multicast Protocol for Ad-Hoc Networks

Mohamed Younis, *Senior Member, IEEE*, Osama Farrag, *Senior Member, IEEE*, and Bryan Althouse

***Abstract*—Ad-hoc networks are becoming an effective tool for many mission critical applications such as troop coordination in a combat field, situational awareness, etc. These applications are characterized by the hostile environment that they serve in and by the multicast-style of communication traffic. Therefore, authenticating the source and ensuring the integrity of the message traffic become a fundamental requirement for the operation and management of the network. However, the limited computation and communication resources, the large scale deployment and the unguaranteed connectivity to trusted authorities make known solutions for wired and single-hop wireless networks inappropriate. This paper presents a new Tiered Authentication scheme for Multicast traffic (TAM) for large scale dense ad-hoc networks. TAM combines the advantages of the time asymmetry and the secret information asymmetry paradigms and exploits network clustering to reduce overhead and ensure scalability. Multicast traffic within a cluster employs a one-way hash function chain in order to authenticate the message source. Cross-cluster multicast traffic includes message authentication codes (MACs) that are based on a set of keys. Each cluster uses a unique subset of keys to look for its distinct combination of valid MACs in the message in order to authenticate the source. The simulation and analytical results demonstrate the performance advantage of TAM in terms of bandwidth overhead and delivery delay.**

***Index Terms*—Multicast communications, message authentication, ad-hoc networks.**

I. Introduction

THE continual advancement in wireless technologies has enabled networked-solutions for many nonconventional civil and military applications. In recent years ad-hoc networks have been attracting increased attention from the research and engineering community, motivated by applications like digital battlefield, asset tracking, air-borne safety, situational awareness, and border protection [1]. In these network applications, it is important to devise efficient network management solutions suitable for nodes that are constrained in onboard energy and in their computation and communication capacities. In addition, the solutions must be scalable to support networks covering vast areas with a large set of nodes that communicate over many hops. These characteristics make the design and management of ad-hoc networks significantly challenging in comparison to contemporary networks. In addition, the great flexibility of ad-hoc networking comes at the price of an increased vulnerability to security attacks and trade-off would be unavoidable at the level of network management and services [2].

Group communication is considered a critical service in ad-hoc networks due to their inherently collaborative operations, where the nodes cooperate in network management and strive to accomplish common missions autonomously in highly unpredictable environment without reliance on infrastructure equipment. For example, in combat missions troops report their status and share observed data in order to become aware of the overall situation and coordinate their actions. In addition, it is common for ad-hoc networks to rely on multicast for management-related control traffic such as neighbor/route discovery to setup multi-hop paths, the establishment of time synchronization, etc. Such multicast traffic among the nodes has to be delivered in a secure and trusted manner. In particular the provided network services need to achieve the following security goals: (1) Confidentiality, to prevent adversaries from reading transmitted data, (2) Message integrity, to prevent tampering with transmitted messages, and (3) Source Authentication, to prevent man-in-the-middle attacks that may replay transmitted data for node impersonation. Confidentiality is achieved by encrypting the transmitted data. The work presented in this paper aims at addressing the second and third goals. Providing an efficient multicast message and source authentication security service that can easily scale for large networks is an important capability for the operation and management of the underlying network.

Source and message authentication is the corroboration that a message has not been changed and the sender of a message is as claimed to be. This can be done by sending a (1) Cryptographic digital signature, or (2) Message Authentication Code (MAC) [3]. The first involves asymmetric cryptography and often needs heavy computation both at the sender and the receiver. The latter involves creating a message and source specific MAC that can be verified by the receiver. Thus, the MAC implicitly ensures message and source integrity. In unicast, a shared secret key is used for MAC generation. Unfortunately, the use of a single shared key in multicast makes the group vulnerable to source impersonation by a compromised receiver. Dealing with multicast as a set of unicast transmissions each with a unique shared key is the most inefficient approach for addressing this concern. These issues combined with other constraints have made contemporary message and source authentication schemes used for multicast traffic in wired and single-hop wireless networks unsuitable for ad-hoc networks.

Manuscript received April 11, 2011; revised August 10, 2011. The associate editor coordinating the review of this paper and approving it for publication was I.-R. Chen.

M. Younis and B. Althouse are with the Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, MD 21250 USA (e-mail: younis@cs.umbc.edu; bkalthouse@gmail.com).

O. Farrag is with Johns Hopkins University, Applied Physics Lab, Laurel, MD 20723 USA (e-mail: osama.farrag@jhuapl.edu).

Digital Object Identifier 10.1109/TNSM.2011.113011.100139

A. Challenges and Design Goals

Multiple factors make multicast authentication in ad-hoc networks very challenging. The issues are fundamentally due to the resource constraints and the wireless links. First, nodes have limited computing, bandwidth, and energy resources which make the overhead of basic asymmetric key-pair cryptography methods very expensive. In addition, the unstable wireless links due to radio interference cause frequent packet loss errors and require a security solution that can tolerate missed packets, as well as differentiate between packet retransmission and replay. Furthermore, the instability of the wireless links makes it unwise to rely on the continual involvement of a trusted authority in the generation and sharing of session keys since a stable connection cannot be guaranteed. On the other hand, while basic symmetric key cryptography methods are efficient, they are ineffective for multicast traffic patterns; since using a common key for all receivers will make it relatively easy to impersonate a sender by any of the receiving nodes.

In addition to being resource efficient and robust to packet loss, a security solution should scale for large group of receivers and long multi-hop paths. Thus, a solution that is based on a distinct authentication key for every receiver will introduce prohibitive overhead to the message and consume significant portion of the available bandwidth. Moreover, the solution should scale for large number of senders by requiring reasonable memory resources at the individual receivers for storing authentication keys. Finally, it is desired to enable the validation of every packet without excessive delay and independent of the other packets. This goal would affect when the authentication code of a packet will be sent and how sensitive the security scheme will be to an occasional delay or a loss of some packets. The motive is that some data may be urgent, e.g. a report on an enemy tank, and should be acted upon as soon as possible, and thus the authenticity of the source should be verified rapidly.

B. Contribution and Organization

This paper proposes a new Tiered Authentication scheme for Multicast traffic (TAM) for ad-hoc networks. TAM exploits network clustering in order to cut overhead and ensure scalability. Multicast traffic within the same cluster employs one-way hash chains to authenticate the message source. The authentication code is appended to the message body. However, the authentication key is revealed after the message is delivered. The idea is similar to the Timed Efficient Stream Loss-tolerant Authentication (TESLA) system [4]. The relatively small-sized cluster would make it possible to keep the nodes synchronized and address the maximum variance in forwarding delay issue of message authentication within a cluster. On the other hand, cross-cluster multicast traffic includes message authentication codes (MACs) that are based on multiple keys. Each cluster looks for a distinct combination of MACs in the message in order to authenticate the source. The source generates the keys at the time of establishing the multicast session. The keys will be securely transmitted to the head of every cluster that hosts one or multiple receivers. The multicast message is then transmitted to the cluster-heads which authenticate the source and then deliver the message to the intended receivers using the intra-cluster authentication scheme. TAM thus combines the advantages of the secret information asymmetry and the time asymmetry paradigms. The analytical and numerical results demonstrate the performance advantage of TAM

The paper is organized as follows. The next section covers the related work. The assumed system model is discussed in Section III. In Section IV, the proposed TAM approach is described in detail. Section V analyzes the performance of TAM and derives bounds for the best and worst case scenarios. The effects on the various parameters on the analytical performance estimates are discussed in Section VI. Section VII reports on the simulation validation of TAM and presents the performance observed in the experiments. Finally, Section VII concludes the paper.

II. Related Work

Source authentication schemes found in the literature can be classified into three categories: (1) secret information asymmetry, (2) time asymmetry, and (3) hybrid asymmetry [3]. The asymmetry property denotes that a receiver can verify the message origin using the MAC in a packet without knowing how to generate the MAC. This property is the key for preventing impersonation of data sources. In secret information asymmetry every node is assigned a share in a secret, e.g., a set of keys. A source appends MACs for the multicast keys so that a receiver verifies the authenticity of the message without being able to forge the MACs for the other nodes [5], [6]. The challenge in using this category of approaches is striking the balance between collusion resilience and performance impact. While the use of a distinct MAC per node imposes prohibitive bandwidth overhead, relying on the uniqueness of the key combinations risks susceptibility to node collusion. TAM pursues secret information asymmetry for its inter-cluster operation and limits the key pool size to suit only the number of clusters. While the description of TAM in Section IV assumes the use of [5], other schemes are equally applicable.

The main idea behind time asymmetry is to tie the validity of the MAC to a specific duration so that a forged packet can be discarded. One-way hash chains are usually employed to generate a series of keys so that a receiver can verify the current key based on an old key without being able to guess the future key. Initially, a source picks a key K0 and generates a chain of keys by recursively applying a one-way hashing function. These keys are used to form the MAC for the individual data packets. The source then reveals the last key, K_l, in the chain to all receivers to serve as the baseline for verification. The key which is used to generate the MAC of a packet is revealed after some time period so that the key cannot be used to impersonate the source. When revealed, the receiver validates the key using K_l or any of the previously revealed keys. TESLA [4] is a very popular example of this category. One of the most distinct advantages of time asymmetry is the minimal per packet overhead that they impose. However, it requires clock synchronization among the communicating parties in order to prevent accepting forged packets, or discarding authentic packets. In addition, in large networks, variations in forwarding delay will force the node to limit the packet transmission rate to avoid revealing next keys to intermediate nodes before all receivers get all previously transmitted packets. These shortcomings limit the scalability of these approaches for multi-hop networks where the maximum end-to-end delay varies significantly among receivers over time and space due to congestions and topology dynamics. Although, some attempts have been made to limit the impact of these issues [7], the scalability of time asymmetry approaches is still questionable. TAM handles the scalability challenge by leveraging clusters and controlling the maximum size of the cluster within which time asymmetric schemes are employed.

Few approaches fall in the third category, mixing both secret-information and time asymmetry [8], [9]. Such hybrid methodology opts to overcome the collusion vulnerability of secret information asymmetry and the tardy verification process of time asymmetry. Basically, a large set of keys is used and only a small subset gets involved in generating the MAC of a particular packet. The subset of keys is picked as a function of the message and is revealed in the same packet. Receivers verify the authenticity of the source as soon as the packet arrives. Since over time a receiver can eventually know all keys, the source periodically employs new keys. Unlike TAM, these schemes will not scale when used in multicast sessions with high packet transmission rates.

The hierarchical structure of a network has been exploited for key management in numerous publications [10]. The idea is to use an upper tier as a trusted authority for key generation. Published approaches vary in selecting the private key generator among the nodes in the higher tier and on the level of coordination among these nodes. Some use ID-based threshold system in order to provision resilience to a node compromise in the upper tier [11], [12], while others use key subsets that are distributed among multiple nodes [13], [14]. TAM relies on the source node in generating the authentication keys for the multicast session; however for generating and distributing keys and establishing of the network hierarchy, including forming clusters and maintaining cluster membership, any of the above hierarchal schemes can be applied to establish trust among nodes.

Some prior work pursued two-tiered protocols to achieve the same design goals of TAM. For example, in [15] monitoring the traffic within a cluster is used by the member nodes to build mutual trust that is considered, along with public certificates, sufficient for authenticating the source of a transmission. For inter-cluster communication public key certifications are used to find out the trust level of the source. The receiver asks a number of introducers within the source cluster to provide the certificate for the source and to share their assessment of its trust level. The introducers sign their reply messages using their private keys to make the certificate valid. Given the overhead for public key cryptography, this approach obviously does not scale well for large multicast groups. In addition, a node that served on a multicast group cannot be virtually evicted from that group without avoiding it while routing the multicast traffic.

Meanwhile, Lu et al. [16] have proposed a message authentication protocol called GSA that opts to efficiently deal with dynamic changes in the topology in a vehicular network. GSA leverages the fact that in some applications, e.g., military squads, vehicles can be naturally grouped due to shared movement pattern. Each group is pre-assigned a leader to act as a trust authority. The group leader is responsible for multicasting commands to group members and interfacing its group to other groups in the network. GSA uses the group attributes to generate a key for authenticating intra-group message traffic. TESLA is applied for inter-group authentication. Like TAM, GSA adopts different security schemes for intra and inter-cluster communication. However, GSA fails to overcome the fundamental delay limitations on TESLA and would not scale to large groups. In addition, GSA does not allow a source to pick a subset of the network nodes in the multicast group and implicitly assumes broadcast. Furthermore, the intra-group operation is vulnerable since the motion characteristics may not be distinct and may overlap among neighboring groups.

III. System Model

A. Architectural Model

An ad-hoc network is a collection of autonomous nodes that together set up a topology without the support of a physical networking infrastructure. Depending on the applications, an ad-hoc network may include up to a few hundreds or even a thousand nodes. Communications among nodes are via multi-hop routes using omni directional wireless broadcasts with limited transmission range. In the system model considered in this paper, nodes are grouped into clusters. The clusters formation can be based on location and radio connectivity [17], [18]. It is assumed that clusters are established securely by using pre-distributed public keys [10], employing a robust trust model [15], [19], or applying identity based asymmetric key-pair cryptographic methods [11], and that a proper key-management protocol is followed in order to perform re-clustering when needed [20], [21]. Clustering is a popular architectural mechanism for enabling scalability of network management functions. It has been shown that clustered network topologies better support routing of multicast traffic and the performance gain dominates the overhead of creating and maintaining the clusters [22], [23].

Each cluster is controlled by a cluster-head, which is reachable to all nodes in its cluster, either directly or over multi-hop paths. Fig. 1 shows an articulation of an example clustered network. Nodes that have links to peers in other clusters would serve as gateways. The presence of gateways between two clusters implies that the heads of these clusters are reachable to each other over multi-hop path and that these two clusters are considered neighbors. If a node moves out its current cluster and joins another, it is assumed that the associated cluster-heads will conduct a handoff to update each other about the change in membership of their clusters; other cluster-heads will not be involved in the handoff events outside their clusters. Mobility is not the focus of this paper; however, prior studies have shown that clustering is advantageous for multicast routing in mobile environments [23].

B. Trust and Threat Models

Nodes are assumed to have public key certificates or assigned identity-based asymmetric keys generated by a common trusted authority. These public keys can be used to form

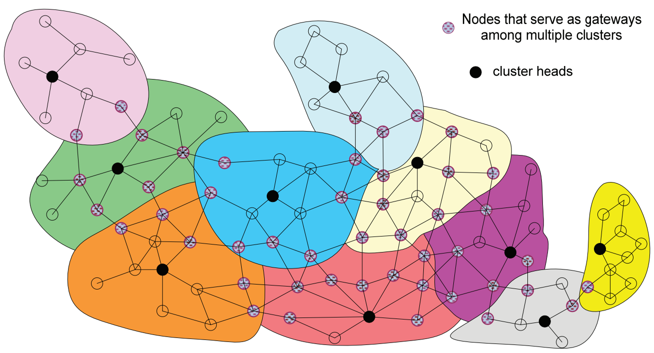


Fig. 1. An example clustered ad-hoc network where each node is reachable to its cluster head via at most 1-hop (2-hop clustering. Nodes that have links to other clusters serve as gateways.

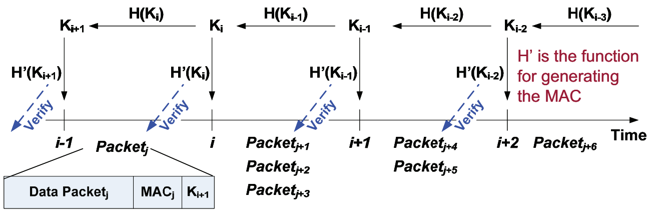


Fig. 2. A source used a key K_i during period j and reveals it in period $j + 1$. thus, a packet in period j will have a MAC based on K_i and will also include $K_i + 1$ for authenticating the packet received in period $j - 1$.

clusters securely [20], [21] and bootstrap TAM. Alternatively, if public key certificates are not suitable, TAM may employ a robust technique to bootstrap mutual trust among the individual nodes [24]. We aim to eliminate any need for interaction with the authority to retrieve the public key of some nodes in the network. TAM bootstrapping will be needed at the time sessions are established and during the formation of a new cluster. Basically, as detailed in Section IV, the source uses asymmetric cryptography to deliver the session keys to the main players in the authentication process. All nodes are to be preloaded with a known one-way hash cryptographic function. The function should be proven secure with extremely low probability that an adversary can determine the input to the function given its output.

This paper mainly considers an adversary who tries to manipulate the system through capturing and compromising some nodes. When a node is captured, its memory can be read or tampered with. Therefore, an adversary would know the keys of a compromised node. In addition, the operation of a compromised node may be manipulated to launch attacks such as replay, impersonation, etc. [2]. TAM opts to ensure source and message authentication in order to counter modify, replay and impersonation attacks. Other attacks are beyond the scope of this paper.

IV. Tiered Authentication of Multicast Traffic

TAM pursues a two-tier process for authenticating multicast traffic in ad-hoc networks. TAM uses clustering to partition a network, and then authenticates multicast traffic by employing time asymmetry for intra-cluster traffic and secret information asymmetry for inter-cluster traffic. As mentioned earlier, clustering is a popular scheme for supporting scalable network operation and management. Several studies have shown that the gains achieved by clustering supersede the overheard in forming and maintain the clusters [22], [23]. TAM leverages such a network management scheme. TAM is explained in the balance of this section.

A. Intra-cluster Source Authentication

Grouping nodes into clusters enables having a reasonably tight bound on the end-to-end delay of packet delivery and will thus enable the use of a time asymmetry based authentication scheme. Intra-cluster authentication in TAM is based on TESLA [4]. Inter-cluster multicast traffic will be authenticated differently as explained below. A source node generates a chain of one-time-use keys using the hash function, e.g., MD5, SHA-1, etc., and shares only that last generated key, K_l, with the receivers. A message can be authenticated only when the used key in the chain is revealed. Fig. 2 demonstrates the authentication process. To verify the authentication key, the receiver recursively applies the cryptographic hash function until reaching K_l. In reality, the receiver can stop when reaching a key that has been used before. A key cannot be used outside its designated time interval and the message will be ignored if the MAC is based on an expired key. Consequently, clock synchronization is required to make sure that the source and destination have the same time reference for key expiration. Therefore, TAM favors small cluster diameters as will be shown shortly. The approach has two distinct advantages, namely:

- The MAC overhead is small; basically a single MAC is used per every multicast packet for all receivers.
- A missed key in a lost packet would not obstruct the authentication process since a receiver can refer back to K_l.

The size of the time interval, which determines when a key is revealed, depends on the clock jitter among nodes in the cluster and on the maximum end-to-end delay between a sender and receivers. Uncertainty about these factors causes the source to be extra conservative in revealing the keys and it thus slows down the data transmission rate. Basically, the receiver will not be able to authenticate the packet contents until the key is transmitted in a later packet, as shown in Fig. 2. The authentication delay may be unacceptable for the application. Perrig et al., [4] have proposed the use of multiple chains in order to expedite the authentication process for close nodes without waiting until further nodes, that are reachable over congested paths, receive the packet.

In TAM, the concern about the authentication delay is generally addressed by the fact that the cluster includes just a subset of the network nodes. The maximum end-to-end delay experienced by an intra-cluster multicast will be mostly dependent on the cluster radius. By controlling the radius of the cluster at the time of cluster formation, i.e., deciding the distance in terms of the number of hops between a member node and the cluster-head [5], [18], it will be possible to tackle this issue. Furthermore, clustering will make it more feasible to synchronize the clock of the nodes in the cluster with some reasonable accuracy. It is well known that for distributed

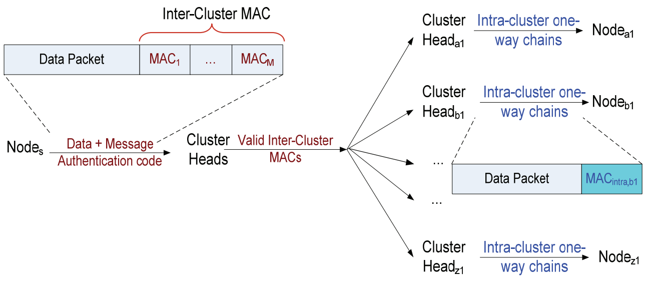


Fig. 3. Illlustrating the steps and packet contents when a node "s" multicast a data packet to nodes "$a1$", "$b1$",$\cdots$,"$z1$" according to TAM.

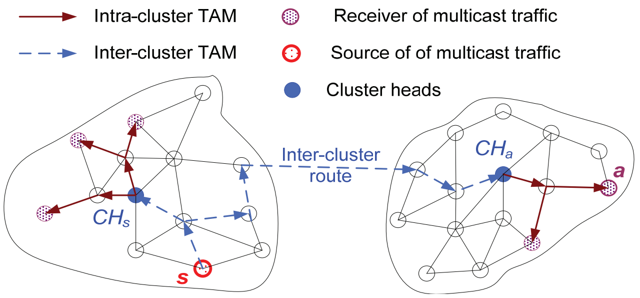


Fig. 4. Summary of the TAM inter-cluster operation. Delivery of the multicast message from a source "s" to all cluster heads applying the TAM inter-cluster authentication, and from each cluster-head, of the designation clusters CH_s and CH_a to the target node "a" apply the TAM intra-cluster protocol.

clock synchronization schemes the accuracy diminishes with increased node population [25]. However, the size of the cluster affect the overhead of the inter-cluster authentication protocol of TAM and will thus be subject to trade-off as explained next. The analytical and simulation results presented in Section VI and VII, respectively, will study the effect of the cluster radius on the delay.

B. *Inter-Cluster Authentication*

Authentication based on time asymmetry requires clock synchronization and thus does not suit large networks. For inter-cluster multicast traffic, TAM applies a strategy based on secret information asymmetry and engages the cluster-heads in the authentication process. Basically, the source "s" that belongs to $Cluster_i$ will send the multicast packets to the heads of all clusters that have designated receivers. For example, if the members of the multicast group for s are residing in clusters g, h, j, and k, node s sends the message to CH_g, CH_h, CH_j, and CH_k. These cluster heads will then forward the message to the receivers in their respective clusters. The rationale is that the MAC will be associated with the cluster rather than the nodes and thus the overhead is reduced significantly. In other words, the multicast from s consists of multiple multicasts; (1) from s to all relevant cluster heads, (2) a distinct multicast within each of the target clusters to relay the message to designated receivers. This can also be advantageous if node mobility is to be dealt with. A node that switches from one cluster to another would only introduce local changes and would not require special handling by the source with respect to the authentication process.

The process goes as follows. The source will generate a pool of M keys. Each of the N_{CL} clusters in the network will be assigned a share L of keys, with $M < L \times\ N_{CL}$. The key share will be sent securely, e.g. using asymmetric cryptographic protocol, to the heads of the individual clusters. The source will then append multiple MACs to the multicast packet; each MAC is based on a distinct key. For a broadcast, exactly M MACs will be included in a packet. The source "s" will then transmit the multicast message to the cluster-heads. Each CH_j checks the MACs and confirm the source authenticity when a set of L MACs in the message are found to be based on the L keys assigned to CH_j by s. The value of M and L is subject to trade-off between security and bandwidth overhead. For $L = 1$, M needs to be equal to N_{CL}.

- Let $K_j^{inter,s}$ be the j^{th} out of the M keys that a source node "s" generates for inter-cluster authentication.
- Let $\mathcal{H}$ be a secure one-way hash cryptographic function
- $K_l^{intra,s}$ is calculated through numberous recursive application of the $\mathcal{H}$ toa root secret key $K_0^{intra,s}$ that cluster head "CH_s" does not reveal. In other words, $K_{l+1}^{intra,s} = \mathcal{H}\left(K_l^{intra,s}\right)$, with $K_l^{intra,s} = \mathcal{H}\left(\mathcal{H}\left(\ldots\left(\left(K_0^{intra,s}\right)\right)\ldots\right)\right)$

1) <u>*Source "s"*</u>:
   - Inter-cluster packet payload

$$P = Data|MAC\left(\text{Data, } K_1^{inter,s}\right), \\ MAC\left(\text{Data, } K_2^{inter,s}\right), \ldots, MAC\left(\text{Data, } K_M^{inter,s}\right)$$

   - Node *"s"* forwards the inter-cluster packet to cluster heads, $\underline{CH}_s$, $\underline{CH}_a$, etx, over an inter-cluster head multicast tree.
2) <u>$\underline{CH}_a$ (Similarly for $\underline{CH}_s$ and other cluster heads:</u>
   <u>- Extract the MAC corresponding to its key share</u> (total of L, e.g., $K_j^{inter,s}, j = 1, \ldots L$
   - Verify MAC$\left(\text{Data, } K_j^{inter,a}\right) \forall j = 1, \ldots, L$
   - Packet$_a$ =Data|MAC$\left(\text{Data, } K_q^{intra,a}\right) \left| K_{q+1}^{intra,a} \right|$Header
   - CH_a multicast Packet$_a$ to local receivers that are members of the multicast group of the source *"s"*
3) <u>Receiver *"a" in the cluster of $\underline{CH}_a$*:</u>
   - Wait for a packet from CH_a that contains $K_q^{intra,a}$
   - Verify that $K_{q+1}^{intra,a} = \mathcal{H}\left(K_q^{intra,a}\right)$
   - Verify MAC$\left(\text{Data, } K_q^{intra,a}\right)$

Higher values of L allow cutting the overhead by assigning unique key combinations to cluster heads ($M = \text{Log}N_{CL}$), possibly at the expense of having a higher risk of collusions if multiple cluster-heads get captured by an adversary. The assignment of the key shares can be based on random selection of L keys from the key pool or based on a localized scheme that minimizes the probability of collusion [14]. It is worth mentioning that N_{CL} would depend on the cluster radius and the used clustering algorithm. The performance of the single key per cluster versus the use of MAC combinations will be studied in Section VI using an analytical estimate of N_{CL}.

Fig. 3 illustrates how TAM handles inter-cluster multicast traffic. The multicast group of a source node "s" includes nodes "$a1$", "$b1$", …, "$z1$". First, node "s" prepares a MAC corresponding to every cluster targeted by the multicast and appends these MACs to the data packet. The source node then forwards the packet to CH_{a1}, CH_{b1}, …, CH_{z1}. Each of the receiving cluster-heads will authenticate the packet using their key share that they got from "s" at the time the multicast session was established. After authenticating the source, each cluster-head forwards the message to the members of the multicast group within its cluster. TAM intra-cluster authentication procedure will be followed inside each cluster, i.e., CH_{a1} will replace the inter-cluster MACs with an intra-cluster time-asymmetry based MAC produced so that receivers like $a1$ can authenticate CH_{a1}, and similarly for CH_{b1}, …, CH_{z1}. Fig. 4 summarizes the inter-cluster procedure and implicitly illustrates the intra-cluster authentication process.

Again it is important to point out the high cost, in terms of bandwidth and power consumption, associated with signing every packet using asymmetric keys. That is why public/private key pairs are used to establish initial trust. Even in unicast sessions the two peers never use asymmetric keys to sign traffic streams, they only use them once to pass a common shared secret, and then the unicast packets are signed using such shared secret. TAM uses asymmetric keys for cluster heads to establish trust with the source and get unique subset of authentication keys for the cluster. In addition, at time of joining a cluster a new node must establish trust with the cluster head in order to ensure that the revealed keys were valid; this needs to be done only one time, and once that trust is established the new node in the cluster can verify subsequent multicast packets because it now can apply the one-way hash function as explained above. Also, with asymmetric keys the multicast message ought to be signed by the key of each node in the multicast groups which obviously does not scale as we have pointed out in Section II.

V. Performance Analysis

The following analysis assumes a network of N nodes with a source is conducting a broadcast, i.e., sending a packet to all the other $(N-1)$ nodes. The focus is on the scalability property assessed based on the bandwidth and delay. The best and worst case performance are analyzed.

A. Baseline Performance

The baseline of comparison is a multicast over a flat network topology. As pointed out in Section II, time asymmetry can introduce vulnerability for large networks unless the data are generated at a very slow rate and excessive delivery delay can be tolerated. Thus, time symmetry alone cannot be practical for networks with large and dynamic multicast groups. Given such vulnerability and the focus of the analysis on the scalability property of TAM, we have deemed it useless to pursue time asymmetric schemes as baseline for comparison. The flat multicast approach mimics the secret information asymmetry and thus more or less captures the fundamental idea of relevant previous work.

In a flat topology the worst performance in terms of the number of transmissions corresponds to having a linear spanning tree of all N nodes. A linear spanning tree in this context refers to the case in which the node degree of the individual nodes is ≤ 2 and a path from a source to a receiver may have to include all the other $N-2$ nodes, i.e., requiring $(N-1)$ transmissions. These transmissions are also sequential with an additive delay i.e., no simultaneity. On the other hand, the best-case performance is achieved over a balanced tree. Assume d is the maximum degree a node in the network. The ideal multicast tree will then be a balanced tree of degree equals d rooted at the source node. Assuming that it is feasible to construct such a balanced multicast tree, the number of transmissions will depend on the height h of the tree. In general, there are d nodes in the first level, d^2 nodes in the second level, etc. Thus,

Number of nodes in a balanced tree of degree $d =$

$$1 + d + d^2 + \ldots + d^h = \frac{d^{h+1}-1}{d-1} \tag{1}$$

The height h of the tree depends on N. To calculate h, assume that N is a perfect match and the leaves of the tree are populated. Thus,

$$N = \frac{d^{h+1}-1}{d-1}, \rightarrow (d-1)\,N = d^{h+1} = (d-1)\,N + 1$$

Taking the logarithms for both sides leads to:

$$h = \log_d\left[(d-1)\,N + 1\right] - 1,$$

Generally if N is not a perfect match,

$$h = \left\lceil \log_d\left[(d-1)\,N + 1\right] - 1 \right\rceil \tag{2}$$

On the multicast tree, the root and the nodes on all levels except the leaves would transmit. Leaves either belong to level h or to level $(h-1)$ if N is not a perfect match. Thus, the number of transmissions in a flat topology would include nodes on levels 0 to $(h-2)$ and only the nodes on level $(h-1)$ that have children.

Transmit $[0, \ldots, (h-2)] = 1+d+d^2+\ldots+d^{h-2} = \frac{d^{h-1}-1}{d-1}$
Transmit [Level $(h-1)$ with children] $= \left\lceil \frac{1}{d}\left(N - \frac{d^h-1}{d-1}\right)\right\rceil$

$$Transmit_{flat} = \frac{d^{h-1}-1}{d-1} + \left\lceil \frac{1}{d}\left(N - \frac{d^h-1}{d-1}\right)\right\rceil \tag{3}$$

The authentication overhead will be the cumulative size of the MACs appended to the message. For information asymmetry, the number of MACs in a broadcast will be $N-1$. For a MAC size of B bits, the overhead will be:

$$\textit{Size overhead per packet} = \#MAC \times B \text{ bits} \tag{4}$$

Finally, the bandwidth overhead for R packet/sec multicast is:
$BW\text{-}overhead_{flat} = packet\ rate \times overhead/packet \times \#\ Trans.$

$$= R \times B \times \#MAC \times Transmit_{flat}$$
$$= R \times B \times (N-1) \times \left(\frac{d^{h-1}-1}{d-1} + \left\lceil \frac{1}{d}\left(N - \frac{d^h-1}{d-1}\right)\right\rceil\right) \tag{5}$$

It is worth noting that a variant of (5) can be pursued when every node is assigned a distinct combination of keys, rather than a unique key per node. In that case, only $\lceil \log_2 N \rceil$ MACs per packet are required and the bandwidth overhead will be:

$$BW\text{-}overhead_{flat} = R \times B \times \lceil \log_2 N \rceil \times \left(\frac{d^{h-1}-1}{d-1} + \left\lceil \frac{1}{d}\left(N - \frac{d^h-1}{d-1}\right)\right\rceil \right),$$
$$\text{where } h = \lceil \log_d \lceil (d-1)\,N + 1 \rceil - 1 \rceil \tag{6}$$

Assuming no medium access collisions among the nodes on the tree, the overall delay in that case is proportional to the tree height h. Let Δ be the time to transmit a bit. Thus, the time for completing the multicast is:

$$Overall\text{-}delay_{flat} = \#\ levels\ in\ tree \times time\ to\ transmit\ a\ packet$$
$$= \#levels\ in\ tree \times packet\ size \times time\ to\ transmit\ a\ bit$$
$$= h \times (Data + B \times \lceil \log_2 N \rceil) \times \Delta \tag{7}$$

Meanwhile, the worst-case overhead and delay corresponds to a linear spanning tree and they are:

$$Worst\text{-}BW\text{-}overhead_{flat}$$
$$= R \times (N-1) \times B \times \lceil \log_2 N \rceil \tag{8}$$
$$Worst\text{-}delay_{flat}$$
$$= (N-1) \times (Data + (B \times \lceil \log_2 N \rceil)) \times \Delta \tag{9}$$

B. Analysis for TAM

In TAM, the multicast involves distinct procedures for intra and inter-cluster operations. For the intra-cluster multicast, the cluster head forwards the packet over a tree and employs a time asymmetry based authentication protocol that requires only a single MAC per packet. Again assuming d-balance tree, the bandwidth overhead can be calculated in a similar manner to the baseline approach above with the exception that the number of nodes is a fraction of the network population and the fact that the bit overhead per packet is much smaller. For a multicast that extends outside the source's cluster, an inter-cluster procedure is invoked to deliver the packet to the cluster heads of the participating receivers. Each cluster-head will then locally multicast the packet within its cluster. Thus, the number of transmissions is the sum of all local (intra-cluster) multicasts inside the individual clusters and the multicast from the source node to the other cluster-heads in the network.

Before deriving the equation, estimates of the number of clusters N_{ch} and the size of the node population per cluster N_c are needed. For that we refer to the analysis of [5], which answers a number of important questions about the properties of k-hop clustering. The following highlights the subset that is relevant to TAM. Two options are discussed based on the assumption on the deployment area and the clustering process.

- *Random Cluster-head Selection*: In a homogeneous network in which no node is pre-designated as a cluster-head, the clustering process is sometimes based on a randomized and distributed procedure. Basically, every node nominates itself as a cluster-head with a probability p. Therefore,

$$The\ average\ number\ of\ clusters\ N_{ch} = \lceil 1/p \rceil \tag{10}$$

$$The\ average\ number\ of\ a\ cluster\ N_c = \lceil N \times p \rceil \tag{11}$$

- *Dominating Set Based Clustering*: If the nodes are uniformly distributed in a square shaped deployment area, i.e., $L \times L$, the average node degree d_{avg} can be calculated using the following formula [26]:

$$d_{avg} = \frac{N\pi T_r^2}{L^2},\ \text{where } T_r \text{ is the node's radio range} \tag{12}$$

Based on [5], the average number of nodes in a k-hop cluster can be estimated as follows. The probability P_c that a node is in a cluster equals the probability that it is at most k hops away of a cluster-head, i.e., less than a distance $k \times T_r$ away from the cluster-head. In other words,

$$P_c = \frac{\pi k^2 T_r^2}{L^2} \tag{13}$$

Thus, N_c and N_{ch} can be further simplified using (12) and (13) as:

$$P_c = \lceil N \times P_c \rceil = N\frac{\pi k^2 T_r^2}{L^2} = k^2 d_{avg} \tag{14}$$

The average number of clusters in that case is:

$$N_{ch} = \left\lceil \frac{\text{Deployment area}}{\text{Cluster coverage}} \right\rceil = \left\lceil \frac{L^2}{\pi k^2 T_r^2} \right\rceil = \left\lceil \frac{1}{P_c} \right\rceil = \left\lceil \frac{N}{k^2 d_{avg}} \right\rceil \tag{15}$$

Intra-cluster Analysis: In the following the best-case overhead for the intra-cluster multicast in TAM is estimated using equations (3)–(5) above. The height of the intra-cluster d-degree balanced multicast tree equals the cluster radius k, since the intra-cluster traffic always originate from a cluster head. Thus,

$$Transmit_{intra} = \#\ clusters \times \#\ time\ within\ a\ cluster$$
$$= N_{ch} \times \left(\frac{d^{k-1}-1}{d-1} + \left\lceil \frac{1}{d}\left(N_c - \frac{d^k-1}{d-1}\right)\right\rceil \right) \tag{16}$$

$$BW\text{-}overhead_{intra} = packet\ rate \times overhead/packet \times \#\ of\ intra\text{-}cluster\ transmissions = R \times B \times Transmit_{intra}$$

$$= R \times B \times N_{ch} \times \left(\frac{d^{k-1}-1}{d-1} + \left\lceil \frac{1}{d}\left(N_c - \frac{d^k-1}{d-1}\right)\right\rceil \right) \tag{17}$$

Obviously, the cluster diameter cannot exceed the height "h" of the balanced tree in the flat topology:

$$2k \leq \lceil \log_d \lceil (d-1)\,N + 1 \rceil - 1 \rceil \tag{18}$$

Using equation (8), the intra-cluster delay is:

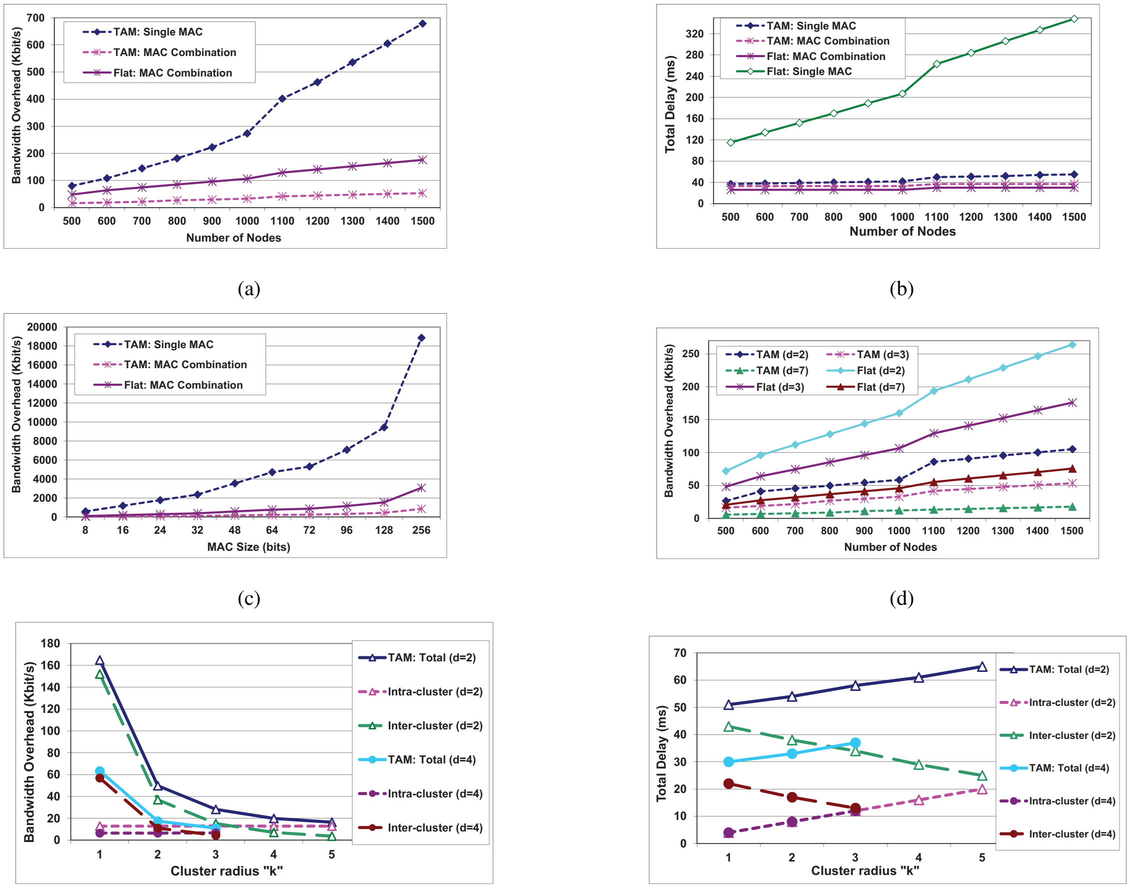
$$Overall\text{-}delay_{intra} = \#\ levels\ in\ tree \times packet\ size \times time\ to\ transmit\ a\ bit$$
$$= k \times (Date + B) \times \Delta \tag{19}$$

Meanwhile, the worst-case intra-cluster overhead and delay correspond to a linear spanning tree and they are:

$$Worst\text{-}BW\text{-}overhead_{intra} = R \times N_{ch} \times (N_c - 1) \times B \tag{20}$$

$$Worst\text{-}delay_{intra} = (N_c - 1) \times (Date + B) \times \Delta \tag{21}$$

Inter-cluster Analysis: Given the k-hop clustering criterion, the distance between the heads of two adjacent clusters is $2k$ hops. For the inter-cluster multicast transmissions, the worst-case scenario is when a linear spanning tree of the source node and all cluster-heads is pursued to route the multicast packet. Thus, we get (22) and (23) at the top of the next page.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 5. Effect of the network size, cluster radius, node degree and MAC size and count on the authentication related bandwidth overhead and the total delivery delay of the multicast packet to all receivers over a d-balanced tree ($k - 2$ in (a)-(d), $d = 3$ for (a)-(c), MAC size = 32 bits for (a)-(b) and (d)-(f), $N = 5000$ for (c), and $N = 800$ for (e)-(f)).

$$Worst\text{-}BW\text{-}overhead_{inter} = R \times (k + 2k\,(N_{ch} - 1)) \times B \times \lceil \log_2 N_{ch} \rceil \tag{22}$$

$$Worst\text{-}delay_{inter} = (k + 2k\,(N_{ch} - 1)) \times (Data + (B \times \lceil \log_2 N_{ch} \rceil)) \times \Delta \tag{23}$$

However, in reality the average case performance will differ significantly since the distance between the heads of neighboring clusters may be less than $2k$. If a balanced tree is formed for inter-cluster multicast, the height of such a tree will be approximately $(h-k)$, where $h = \lceil \log_d \lceil (d-1)\,N + 1 \rceil - 1 \rceil$ as defined in (2). The height is based on the intuition that forming a balanced tree rooted at CH_i can cover all nodes in the network, including non-CH members of all other clusters, over h hops in the worst-case. Since the radius of a cluster is k and assuming all other cluster-heads are at the center of their respective clusters, CH_i would be able to reach the furthest cluster head CH_i over $(h-k)$ hops. As explained earlier when deriving equation (3), the number of transmissions over a d-balanced tree of height h consists of all broadcasts to reach all nodes on level $(h-1)$ and transmissions to reach the leaves on level h. For TAM's inter-cluster multicast, the leaves are all cluster heads, other than the sender CH_i, and the inter-cluster transmissions would thus be:

$$Transmit_{inter} = \frac{d^{h-k-1} - 1}{d - 1} + N_{ch} - 1 \tag{24}$$

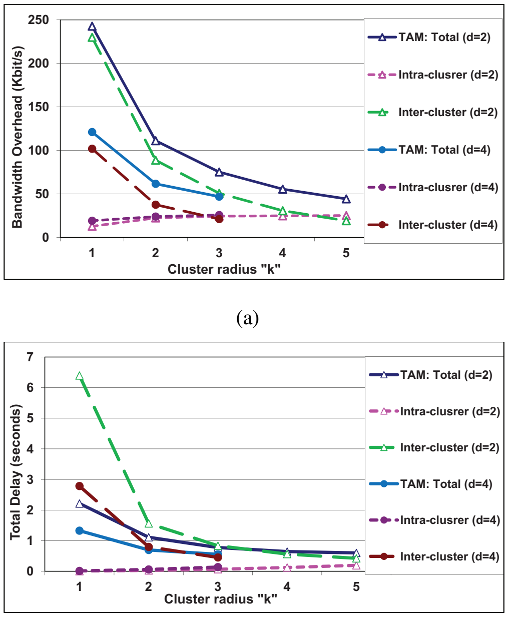
The corresponding bandwidth overhead and delay are:

$BW\text{-}overhead_{inter} = packet\ rate \times overhead/packet\ (MAC\ size)\ times\ Transmit_{inter} = R \times B \times \lceil \log_2 N_{ch} \rceil \times Transmit_{inter}$

$$= R \times B \times \lceil \log_2 N_{ch} \rceil \times \left(\frac{d^{h-k-1} - 1}{d - 1} + N_{ch} \right) \tag{25}$$

$Overall\text{-}delay_{inter} = \#\ levels\ in\ inter\text{-}cluster\ tree \times packet\ size \times time\ to\ transmit\ a\ bit$

$$= (h - k - 1) \times (Data + B \times \lceil \log_2 N_{ch} \rceil) \times \Delta \tag{26}$$



(a)

(b)

Fig. 6. The bandwidth overhead and the total time delay until all multicast packets are delivered as a function of the cluster radius and using a linear spanning tree for routing packets ($N = 800$).

VI. Numerical Results

This section shows numerical results based on the analysis in the previous section. The goal is to articulate the effect of the various parameters on the performance of TAM and highlight how a suitable configuration can be picked. The graphs in Fig. 5 report on the bandwidth overhead imposed by the authentication process and the overall delay for delivering the packets to all receivers. Again these results are for a broadcast, i.e. one source to all other nodes in the network. The varied parameters include the network size, the cluster radius, MAC size and count, and node degree.

The results in Figs. 5 and 6 are based on 4K bit fixed size multicast packets sent once every second over a 1M bit/s radio channel. Unless varied in the graph, the MAC size is set to 32 bits. For Fig. 5, a best-case d-balanced tree is assumed for disseminating the multicast traffic for both TAM and the baseline approach. Fig. 6 is based on a spanning tree, i.e., worst-case performance. Since for spanning tree multicast routing the bandwidth overhead of the flat approach is prohibitive, graphs involving the flat approach are not included and only the effect of relevant parameters on TAM's performance is captured in Fig. 6.

- *Effect of the network size*: Fig. 5(a) demonstrates the performance advantage of TAM in terms of the bandwidth overhead. The graph shows that when using a MAC combination, rather than a single MAC per node, TAM introduces a minimal overhead that slightly grows as the number of nodes increases. However, using a single MAC per node boosts the overhead substantially. The bandwidth overhead for TAM under a single MAC per node is significantly more than the baseline with MAC combinations. This indicates the dominance of the effect of the MAC size on performance. It is worth noting that the single MAC per node imposes prohibitive overhead if a flat topology is pursued and is not shown since the scale of the y-axis will not allow the other curves to be visible. In summary, TAM enables boosting the network resilience to collusions if desired, an option that is practically infeasible for the baseline approach. Fig. 5(b) shows the time until all receivers get the multicast packet. Due to the assumption of a d-balanced tree, the flat topology involves the least delay that increases at a very slow pace when the network grows. The delay for TAM is slightly higher given the multi-step operation, i.e., intra and inter-cluster. Nonetheless, TAM significantly outperforms the baseline for the case of a single MAC per node. When considering both Figs. 5(a) and (b), TAM clearly stands out as a practical approach for collusion resistance authentication.
- *Effect of the MAC size and count*: Fig. 5(c) shows the scalability of TAM with respect to the size of the digital signature. Longer signatures are usually pursued as a means for boosting the cryptographic strength of the security solution. TAM scales very well, even when a single MAC per node is used. The overhead grows almost exponentially for the flat approach and, like Fig. 5(a), is not included in the graph in order to allow the other curves to be readable.
- *Effect of node degree*: Fig. 5(d) captures the effects of the node degree on the performance. The results are based on the use of MAC combinations. In general, increased network connectivity would reduce the size of the multicast tree and decrease the number of transmissions, and consequently the bandwidth overhead. However, TAM yields very distinct performance even with low level of network connectivity.
- *Effect of cluster radius*: Figs. 5(e) & (f), and Fig. 6 study the effect of the cluster radius k in a network of 800 nodes, while using d-balanced and linear spanning multicast trees, respectively. The choice of k is a key design issue for TAM. It is worth noting that for $d = 4$, k cannot exceed 3, per equation (18) above. While the inter-cluster operation imposes the most bandwidth overhead and dominates the delay, the effect diminishes with an increased cluster radius. Meanwhile, the intra-cluster multicast scales very well for large k values with respect to the introduced overhead. However, the intra-cluster delay performance depends on the multicast routing topology. For a d-balanced multicast tree the delay grows when increasing the cluster radius and almost matches the inter-cluster related delay for large k values, as indicated by Fig. 5(f). Nonetheless, the delay grows almost linearly with increased k, and increases in significance under low connectivity conditions, i.e. for small d values. On the other hand, the delay diminishes with the growth of the cluster radius when a linear spanning tree is used for routing the packets (Fig. 6(b)). Considering Figs. 5(e), 5(f), 6(a) and 6(b) indicates that a cluster radius of $2-3$ hops would be a reasonable choice for TAM since it seems to balance the interest in low overhead and short delay and exhibits less dependence of

the multicast routing topology. This point will be revisited in the next section when discussing the average results obtained through simulation.

VII. Simulation Experiments

The numerical results shown above demonstrate the advantages of TAM and characterize the impact of the various parameters on the achievable performance under best case and worst case conditions, e.g. balanced or spanning multicast trees. To study the average TAM performance, simulation experiments have been pursued using NS2 [27]. This section describes the simulation environment and analyzes the observed performance.

A. Simulation Environment

TAM is a generic authentication protocol that relies on link layer, or geo-zoning, clustering to partition the network. In the implementation, we employed the distributed k-hop cluster algorithm of [28] to provide the required link-layer based clustering capability. This clustering approach extends the lowest-ID single-hop clustering algorithm to k-hop. Each node joins the CH with the lowest ID that can be reached over a path of k hops or less, and the distance between two CHs should be $k+1$ hops or more. Although, TAM permits the use of any two tiered multicast routing scheme, we chose to build a TAM simulation model using a tree based two tiered routing scheme in order to easily compare the simulation results to the tree based analytical estimates of best case and worst case topologies covered in the previous section. For this reason, as well as the availability of source code, MAODV has been selected. MAODV [29] is a multicast extension to the commonly used AODV (Ad hoc On-Demand Distance Vector Routing) unicast router.

The NS2-2.26 implementation of MAODV [30] has been ported to operate within the NS2.33 version, which we used to measure TAM performance. We have extended the NS2 MAODV standard model by introducing a bridging function on-top of the basic MADOV protocol in order to provide a two-tiered cluster based forwarding capability. The top tier handles traffic from a source node to each CH and the lower tier forwards traffic from a CH to nodes within its cluster. In the implementation, a TAM multicast session is realized using $(N_{ch}+1)$ MAODV multicast sessions, one for inter-cluster and N_{ch} intra-cluster sessions. Each of the $(N_{ch}+1)$ sessions is assigned a unique multicast group address.

Fig. 7 shows how TAM is integrated in NS2. The NS2 MAODV agent is responsible for the routing of multicast traffic. The CH bridge agent applies TAM's protocol by making each cluster head join the common inter-cluster multicast session, and establish an intra-cluster local multicast session. This enables the enforcement of routing constraints based on TAM intra- and inter-cluster multicast. Standard MAODV routing protocol resources were used for both intra-cluster and inter-cluster routing. Non-CH nodes will only join the multicast session initiated by the local cluster head. CHs have the responsibility of bridging between the inter-cluster and intra-cluster authentication domains. A CH will forward each multicast packet received on its intra-cluster multicast address to the rest of cluster heads by substituting the intra-cluster multicast group destination address with the session inter-cluster multicast group address. Then each CH will forward each multicast packet received on its inter-cluster multicast address to local cluster nodes by substituting the multicast session inter-cluster address with the multicast address assigned to its cluster. This approach allowed us to leverage the standard MAODV protocol without modifications to the tree based MOADV routing approach while meeting TAM's two-tiered routing requirement.

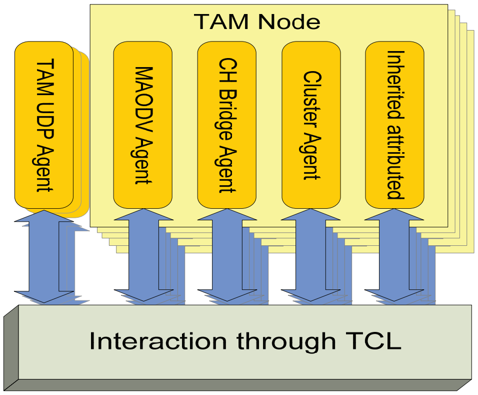


Fig. 7. Illustrating the NS2 implementation of TAM. The basic attributes are inherited from the node calss in NS2. The various agent classes are added to handle functionalities related to TAM.

The primary purpose of the introduced NS-2 TAM UDP Agent is to manipulate the destination multicast address of the multicast traffic source. If the traffic source is a cluster member, the inter-cluster multicast group address is chosen as the destination address. If the traffic source is a CH, then duplicate packets are sent using both inter- and intra-cluster multicast group addresses. Additionally, the TAM UDP Agent includes instrumentation to collect statistics on packet receptions. For flat baseline experiments, we configure nodes with standard MAODV routing and UDP agents.

B. Experiments Setup and Results

Again, the focus of the experiments is to capture the bandwidth overhead and delay metric under TAM compared to a flat approach in a typical network topology. It was observed in tryout experiments that packet loss due to medium access collisions is so major for networks of 400 nodes or more that hardly any meaningful overhead and delay statistics about TAM and the baseline approach could be obtained. In order to isolate the effect of overhead introduced by the authentication methods under evaluation, our simulation results excluded all adverse effects of link layer collisions and retransmissions on overheads and end-to-end delay.

In the experiments, we varied the number of nodes, the size of MAC, the average node degree, and the cluster radius. The results of the individual experiments are averaged over 30 runs using distinct network topologies. All results are subjected to 90% confidence interval analysis and stays within 10% of the sample mean. The simulation results are presented in Fig. 8. To compare with the analytical results, similar parameter settings have been used, namely, 4K bit fixed size multicast

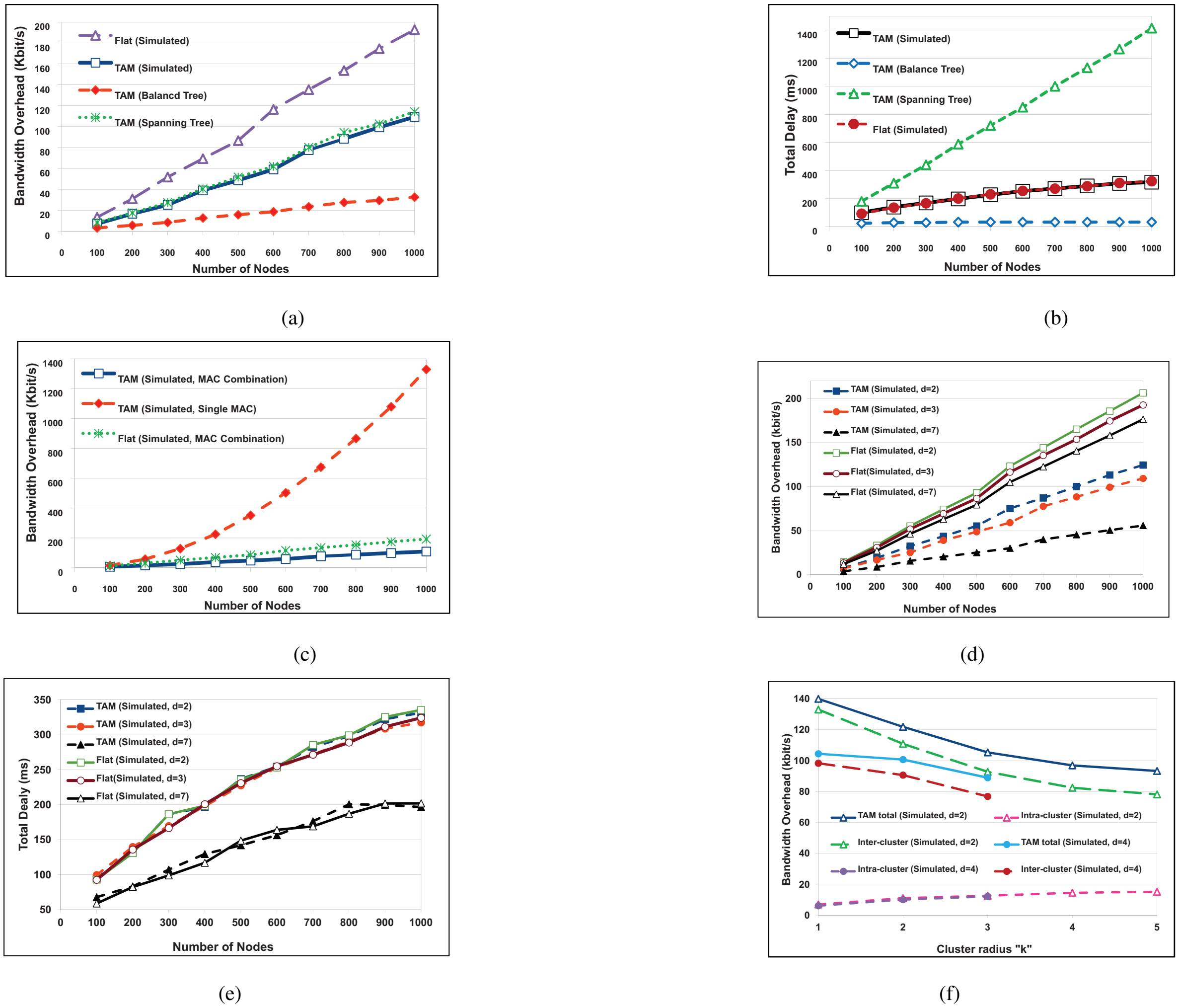
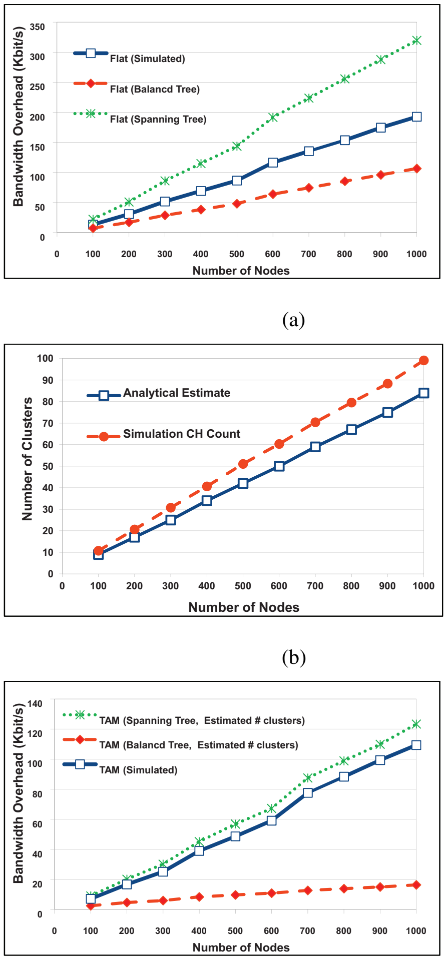


Fig. 8. The observed simulation results while varying the network size, cluster radius, node degree and MAC count on the authentication related bandwidth overhead and the total delay of the multicast packet to all receivers ($k = 2$ and $d = 3$ in (a)-(c), $k = 2$ for (c)-(e), and $N = 800$ for (f)).

packets sent once every second and the MAC size is set to 32 bits unless varied in the experiment. In addition, the results are based on broadcast, where a source is picked at random and the message is then transmitted to all other nodes in the network. MAC combinations are used for achieving information asymmetry in the TAM inter-cluster authentication, unless explicitly mentioned otherwise. The following is the analysis of the obtained results:

- *Effect of the number of nodes*: Figs. 8(a) and 8(b) compare the bandwidth overhead and delay performance of TAM observed in the simulation, to the performance of the flat approach and to that of TAM based on the analytical estimates of Section VI. The cluster radius was set to 2 and the node degree was about 3 when running these simulation experiments. Fig. 8(a) confirms TAM's advantage over the flat baseline. As mentioned earlier, MAODV is used as the underlying routing protocol for both schemes. The performance gap between the flat approach and TAM widens as the network size grows, demonstrating the scalability of TAM. When compared to the balanced-tree based (best-case) performance, the simulation results for TAM are worse than expected. In fact TAM's performance in the simulation is close to the spanning-tree (worst-case) performance. Fig. 9 helps in explaining these results. First, Figure 9(a) compares the simulation results of the flat approach to that of the corresponding best (balanced tree) and worst (spanning tree) cases calculated using the equations in Section VI. Fig. 9(a) indicates that MAODV did not establish a spanning tree for multicast routing with the simulated performance half way between best- and worst-case performances. Thus, one expects that the performance gap between the TAM bandwidth performance to that based on a balanced tree is attributed to a factor other than the multicast routes formed by MAODV. Since the inter-cluster component of TAM uses multiple MACs per packet and thus imposes the most bandwidth overhead, the number of formed clusters is tracked and reported in Fig. 9(b) and the corresponding spanning and balanced tree performances are plotted in Fig. 9(c). Basically, more clusters are formed in the simulation than estimated by the analytical model. Such difference in the number of formed clusters explains the discrepancy in the simulated and analytical TAM performance. Analyzing the plots of Figs. 9(b) and 9(c) indicate that the boosted number of clusters made the inter-cluster operation dominate the overhead



(a)

(b)

(c)

Fig. 9. (a) comparison of the observed performance of the flat approach in the simulation to the analytically-estimated best and worst case, (b) the number of clusters formed for the various network sizes, and (c) the simulation results of TAM bandwidth overhead as compared to analytical estimates based on the formed number of clusters during the simulation. Results are based on $k = 2$ and $d = 3$.

in the simulation. In conclusion, the performance of TAM favors fewer clusters count, and dense and highly connected clusters. Fig. 8(b) reports the delay performance. This delay is due to packet transmissions since the effect of medium access collisions has been muted, in our analysis and simulation experiments, as mentioned earlier. The results indicate that on the average the longest path from a source to members of the multicast group has almost the same number of hops in both TAM and the flat approach. This observation confirms that MOADV is not the main reason for the obtained TAM overhead performance in Fig. 8(a), as concluded above. Also, the delay performance indicates that the routing of multicast traffic in the simulation was not over a spanning tree which is consistent with our analysis of Fig. 9. In general, the simulation results demonstrate that in most deployments, the two-tiered routing requirements for the TAM approach does not introduce additional delay.

- *Effect of the inter-cluster authentication scheme*: TAM pursues information asymmetry for inter-cluster authentication of multicast traffic. Fig. 8(c) compares the bandwidth overhead when including MAC combination or a single MAC per cluster in the individual multicast packets sent to the cluster-heads. The performance is also compared to the MAC-combination based implementation of the flat approach. In essence Fig. 8(c) is the simulation based, i.e., average-case performance, version of Fig. 5(a) which analyzes the best-case performance over a balanced-tree routing topology. The simulation results are very much consistent with Figure 5-(a). The main difference is that going with a single MAC per packet in the simulation seems to grow the overhead at a significantly higher rate than in the analytical estimates when the network population increases. This is attributed to the higher cluster count in the simulation as shown in Fig. 9(b). It is worth noting that the use of a single MAC per packet with the flat approach imposes excessive bandwidth overhead and is not plotted in Fig. 8(c) to make the graph readable.
- *Effect of the node degree*: Fig. 8(d) plots the average bandwidth overhead in the simulation experiments as the network connectivity is varied. The results for both TAM and the flat approach are plotted. When compared to Fig. 5(d), which shows the corresponding graph based on analytical estimates over a balanced multicast tree, we observe that TAM's performance is a bit lower and the effect of the node degree is more noticeable. The performance drop is attributed to the fact that more clusters are formed in the simulation and MOADV is not forming a balanced tree. Nonetheless, TAM still sustains its performance edge over the flat approach. It is worth noting that a significant boost in the connectivity of the network makes a major positive impact on TAM performance, while the flat approach does not benefit as much. The reason is that, for a fixed cluster radius, increasing the node degree grows the cluster sizes and decreases the number of clusters. This cuts on the inter-cluster authentication overhead, which is relatively higher than the intra-cluster overhead. The node degree also has a positive impact on the delivery delay, as shown in Fig. 8(e). This can be attributed to the reduced height of the multicast tree with the increase in network connectivity. As pointed out about Fig. 8(b), the reported delay is the accumulation of the transmission time from the source until the last node receives the multicast packet. The two-tiered forwarding requirement of TAM does not seem to extend the longest path on the multicast tree as compared to the flat approach, as evident by the matched delay results in Fig. 8(e) for the corresponding node degree.
- *Effect of the cluster radius*: Fig. 8(f) reports the bandwidth overhead as a function of the cluster radius. The number of nodes is fixed at 800 in these experiments. The results for $d = 2$ and $d = 4$ are shown to capture the effect of network connectivity as well. The results are consistent with Figs. 5(e) and 6(a), where the increase in the cluster radius lowers the overhead, mainly because of the decrease in the cluster count. It is worth noting that the simulation results falls between

the best case analytical performance estimates (Fig. 5(e)) and the worst case (Fig. 6(a)), which is consistent with earlier observations about the other simulation results. It is important to note that the delay performance in the simulation was not affected by the change in the value of "k". This is explained by the fact that the multicast routes formed by MAODV do not fit d-balanced nor linear spanning trees and thus the total delay would not noticeably grow as in Fig. 5(f) nor diminish as in Figu. 6(b) with the increase in the cluster radius.

Configuring TAM: Both security and performance factors have to be considered when employing TAM. With respect to resilience to impersonation and replay attacks, TAM limits the effect of a node compromise to within a cluster. If a cluster member is captured, the TESLA-based intra-cluster authentication will deem any attempt by an adversary to launch these attacks ineffective. Meanwhile, the vulnerability to these attacks due to the capture of a CH node is still limited to within the cluster since only the key share of the compromised CH node will be uncovered and the adversary will not be able to fool other cluster-heads. However, a compromised CH cannot be prevented from launching impersonation and replay attacks against the members of its own cluster. Although the probability of capturing a CH is significantly low given the low CH count within the node population, it is advisable to have small clusters in order to mitigate the effect when it happens.

As shown above and also in Section VI, the network performance is a major consideration for TAM. While a large value of k seems to be advantageous for TAM's bandwidth overhead and would not cause a concern about the delivery delay, the management complexity of a cluster usually grows significantly with the increase in the cluster radius [28]. In addition, the larger cluster radius will result in an increased intra-cluster delay variability which is not suitable for one way hash chain multicast authentication methods. As such, based on our observations from the analytical and simulation results and the security analysis above, a TAM system with a cluster radius of 2 or 3 would yield a major performance gain over current multicast schemes and strike a balance between the objectives of achieving high network performance and increasing resilience to impersonation and replay attacks.

VIII. Conclusion

In recent years there has been a growing interest in the use of ad-hoc networks in security-sensitive applications such as digital battlefield, situation awareness, and border protection. The collaborative nature of these applications makes multicast traffic very common. Securing such traffic is of great importance, particularly authenticating the source and message to prevent any infiltration attempts by an intruder. Contemporary source authentication schemes found in the literature either introduce excessive overhead or do not scale for large networks. This paper has presented TAM, which pursues a two tired hierarchical strategy combining both time and secret-information asymmetry in order to achieve scalability and resource efficiency. The performance of TAM has been analyzed mathematically and through simulation, confirming its effectiveness. In addition, the effect of the various parameters has been studied and guidelines have been highlighted for picking the most suitable configuration in the context of the particular application requirements; most notably having a cluster radius of 2 or 3 hops appears to be the most suitable for TAM. Our future work plan includes studying the effect of different clustering strategies on the performance of TAM.

Acknowledgment

The authors are indebted to Prof. Paolo Bellavista and Eugenio Magistretti for providing the source code of their k-hop clustering algorithm used the simulation validation of TAM. In addition, the authors would like to thank Fatih Senel for providing a tool that generates strongly connected random network topologies for the NS2 experiments.

References

[1] C. E. Perkins, *Ad Hoc Networking*. Addison-Wesley, 2001.

[2] H. Yang, *et al.*, "Security in mobile ad-hoc wireless networks: challenges and solutions," *IEEE Wireless Commun. Mag.*, vol. 11, no. 1, pp. 1536–1284, Feb. 2004.

[3] Y. Challal, H. Bettahar, and A. Bouabdallah, "A taxonomy of multicast data origin authentication, issues and solutions," *IEEE Commun. Surveys & Tutorials*, vol. 6, no. 3, pp. 34–57, 2004.

[4] A. Perrig, R. Canetti, D. Song, and D. Tygar, "Efficient authentication and signing of multicast streams over lossy channels," in *Proc. 2000 IEEE Symposium Security Privacy*.

[5] R. Canetti *et al.*, "Multicast security: a taxonomy and efficient constructions," in *Proc. 1999 IEEE INFOCOM*.

[6] R. Safavi-Naini and H. Wang, "Multi-receiver authentication codes: models, bounds, constructions, and extensions," *Inf. Computation*, vol. 151, no. 1–2, pp. 148–172, May 1999.

[7] Perrig, *et al.*, "Efficient and secure source authentication for multicast," in *Proc. 2001 Network Distributed System Security Symposium*.

[8] A. Perrig, "The BiBa one-time signature and broadcast authentication protocol," in *Proc. 2001 ACM Conf. Computer Commun. Security*.

[9] L. Reyzin and N. Reyzin, "Better than BiBa: short one-time signatures with fast signing and verifying," in *Proc. 2002 Australian Conf. Info. Security Privacy*, pp. 144–153.

[10] A. M. Hegland, E. Winjum, S. F. Mjolsnes, C. Rong, O. Kure, and P. Spilling, "A survey of key management in ad hoc networks," *IEEE Commun. Surveys & Tutorials*, vol. 8, no. 3, pp. 48–66, Dec. 2006.

[11] F. R. Yu, H. Tang, P. Mason, and F. Wang, "A hierarchical identity based key management scheme in tactical mobile ad hoc networks," *IEEE Trans. Netw. Service Management*, vol. 7, no. 4, pp. 258–267, Dec. 2010.

[12] R. Gennaro, *et al.*, "Strongly-resilient and non-interactive hierarchical key-agreement in MANETs," in *Proc. 2008 European Symp. Research Computer Security*.

[13] G. Hanaoka, T. Nishioka, Y. Zheng, and H. Imai, "A hierarchical non-interactive key-sharing scheme with low memory size and high resistance against collusion attacks," *Computer J.*, vol. 45, no. 3, pp. 293–303, 2002.

[14] M. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 18, pp. 865–882, Aug. 2006.

[15] E. C. H. Ngai and M. R. Lyu, "An authentication service based on trust and clustering in wireless ad hoc networks: description and security evaluation," in *Proc. 2006 IEEE International Conf. Sensor Networks, Ubiquitous, Trustworthy Computing*.

[16] Y. Lu, B. Zhou, F. Jia, and M. Gerla, "Group-based secure source authentication protocol for VANETs," in *Proc. 2010 IEEE GLOBECOM Workshop Heterogeneous, Multi-hop Wireless Mobile Networks*.

[17] M. Youssef, A. Youssef, and M. Younis, "Overlapping multihop clustering for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 12, pp. 1844–1856, Dec. 2009.

[18] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Commun. Surveys & Tutorials*, vol. 1, no. 1, pp. 31–48, 2005.

[19] P. B. Velloso, *et al.*, "Trust management in mobile ad hoc networks using a scalable maturity-based model," *IEEE Trans. Network Service Management*, vol. 7, no. 3, Sep. 2010.

[20] R. Azarderskhsh and A. Reyhani-Masoleh, "Secure clustering and symmetric key establishment in heterogeneous wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, article ID 893592, 2011.

[21] L. Wang and F. Gao, "A secure clustering scheme protocol for MANET," in *Proc. 2010 International Conf. Multimedia Inf. Netw. Security*.

[22] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu, "A survey of multicast routing protocols for mobile ad-hoc networks," *IEEE Commun. Surveys & Tutorials*, vol. 11, no. 1, pp. 78–91, first quarter 2009.

[23] M. Younis, O. Farrag, and S. Lee, "Cluster mesh based multicast routing in MANET: an analytical study," in *Proc. 2011 IEEE International Conf. Commun.*.

[24] D. Balfanz, *et al.*, "Talking to strangers: authentication in ad-hoc wireless networks," in *Proc. 2002 Network Distrib. System Security Symposium*.

[25] K. Marzullo and S. Owicki, "Maintaining the time in a distributed system," in *Proc. 1983 ACM Symposium Principles Distrib. Computing*.

[26] A. Savvides, C. C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. 2001 ACM International Conf. Mobile Computing Netw.*, pp. 166–179.

[27] The Network Simulator - ns-2. Available: http://www.isi.edu/nsnam/ns/

[28] G. Angione, P. Bellavista, A. Corradi, and E. Magistretti, "A k-hop clustering protocol for dense mobile ad-hoc networks," in *Proc. 2006 IEEE International Conf. Distrib. Computing Systems Workshop*.

[29] E. M. Royer and C. Perkins, "Multicast ad-hoc on-demand distance vector (MAODV) routing," Internet Draft, University of California, Charles E. Perkins Nokia Research Center, July 2000.

[30] Y. Zhu and T. Kunz, "MAODV implementation for NS-2.26," Technical Report SCE-04-01, Dept. of Systems and Computing Engineering, Carleton University, Jan. 2004.

Mohamed Younis (M'99–SM'05) received B.S. and M.S. degrees from Alexandria University, Egypt, and the Ph.D. degree in computer science from the New Jersey Institute of Technology, USA. He is currently an Associate Professor in the Department of Computer Science and Electrical Engineering at the University of Maryland Baltimore County (UMBC). Before joining UMBC, he was with the Advanced Systems Technology Group, an Aerospace Electronic Systems R&D organization of Honeywell International, Inc. While at Honeywell, he led multiple projects for building integrated fault tolerant avionics and dependable computing infrastructure. He also participated in the development of the Redundancy Management System, which is a key component of the Vehicle and Mission Computer for NASA's X-33 space launch vehicle. Dr. Younis' technical interest includes network architectures and protocols, wireless sensor networks, embedded systems, fault tolerant computing, secure communication, and distributed real-time systems.

Dr. Younis has five granted and two pending patents. He served as the chair of LCN'10 and program co-chair of LCN'09, the ad-hoc and sensor networks symposium of ICC'09, and the wireless networks symposium of ICC'11. In addition, he serves/served on the editorial board of multiple journals and the organizing and technical program committee of numerous conferences. Dr. Younis has published over 150 technical papers in refereed conferences and journals.

Osama Farrag is a Senior Member of Professional Staff at Johns Hopkins University Applied Physics Lab (JHU/APL). His educational background includes a B.S. in E.E., an M.S. in computer science from Old Dominion University, and an M.S. in technical management from Johns Hopkins University. Prior to his tenure at JHU/APL, Mr. Farrag had over 25+ years of experience in the telecommunication/data communication industry, where he held senior level leadership positions at different organizations that developed and deployed circuit and packet switching equipment and cellular infrastructure products.

Mr. Farrag has authored several patents and publications. His current research interests include MAC layer protocols and cross-layer optimization for mobile ad-hoc networks, secure communication, and trustworthy software and system architectures

Bryan L. Althouse received the M.S. degree in computer engineering from the University of Maryland, Baltimore County, in 2010, and the B.S. degree in electrical engineering from the Georgia Institute of Technology in 1997. He is currently employed as a Lead Engineer at 3Phoenix Inc. (Hanover, MD). He also functions as the lead electrical engineer and acting project manager for the development of a new military sonar array. Before joining 3Phoenix, Mr. Althouse was a Senior Software Engineer for Optinel Systems Inc. (Elkridge, MD) where he developed optical fiber based communications equipment. Prior to that, he was employed as an Engineer by the Naval Research Lab (Washington D.C.) where he designed electronics for optical fiber sensors.