International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Issue 6, Volume 2 (June 2015) www.ijirae.com

A Distinct Authentication Key for TAM Protocol for Ad-hoc Networks

Sadhna Yadav , Bharthi V Kalmath
PG Research Scholar, Department of CSE, SVCE, Bangalore, Karnataka, India
Assistant Professor, Department of CSE, SVCE, Bangalore, Karnataka, India

ABSTARCT-One of the key issues that need to be addressed in wireless sensor network field is how to create a most efficient energy system. Ad-hoc Networks are becoming an effective tool for many mission critical applications such as troop coordination, situational awareness Etc. Ad-hoc Networks having limited computation and communication resources. To unguaranteed connectivity to trust authorities make known solutions for trusted authorities make knows solutions for single hop wireless networks. In this Project I am presenting A Distinct Authentication Key for New TAM Protocol For large scale dense ad-hoc networks. TAM exploits network clusters to reduce overhead and ensure scalability. Multicast traffic with in a cluster employs a "One way hash function" chain in order to authenticate the message source. Cross cluster multicast traffic includes Message Authentication Codes (MAC's) that are based on a set of keys. Result in terms of bandwidth overhead and delivery delay.

Keywords —Multicast Communication, 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. 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. 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].

AIM: 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.

A. CHALLENGES AND DESIGN GOALS

1. A distinct authentication key for every receiver will
Introduce prohibitive overhead to the message and consume
Significant portion of the available bandwidth.
2. Reasonable memory resources at the individual receivers for storing authentication keys.
3. To enable the validation of every packet without excessive delay and independent of the other packets.

B. CONTRIBUTION

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 (Key is revealed after the message is delivered) code is appended to the message body.

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.

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. 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 a 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. 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 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.

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.

III PROPOSED SYSTEM

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.

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. 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.

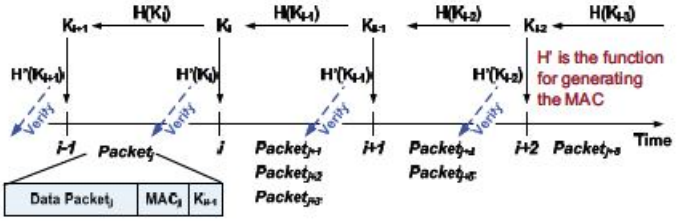

Fig 1: A source used a key K_i during period j and reveals it in period

$j+1$. 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 $j - 1$.

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 K1

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. 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], [15], 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 clock synchronization schemes the accuracy diminishes with increased node population . 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.

B. Inter-Cluster Authentication

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*$_l$ 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}.

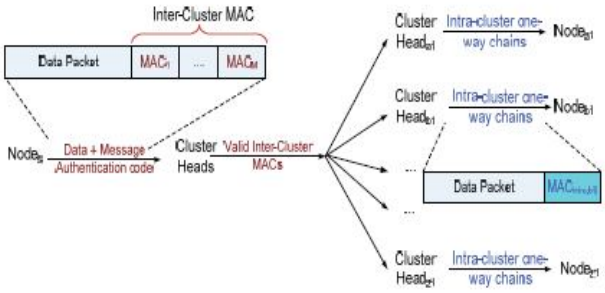

Fig 2: Illustrating the steps and packet contents when a node "*s*" multicast a data packet to nodes "*a*1", "*b*1",· · · ,"*z*1" according to TAM.

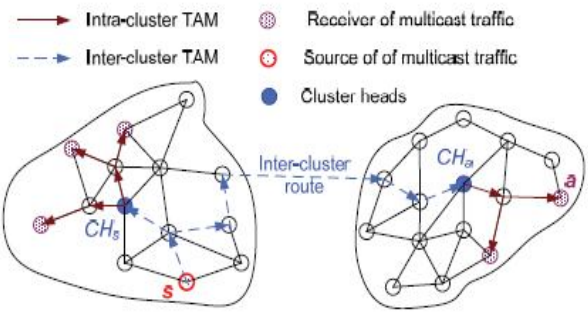

Fig 3: Summary of the TAM inter-cluster operation. Delivery of the multicast message from a source "*s*" to all cluster heads applying the TAM inter-cluster authentication, and from each cluster-head, of the designation clusters CH_s and CH_a to the target node "*a*" apply the TAM intra-cluster protocol.

It is worth mentioning that N_{CL} would depend on the cluster radius and the used clustering algorithm. Fig. 2 illustrates how TAM handles inter-cluster multicast. traffic. The multicast group of a source node "*s*" includes nodes "*a*1", "*b*1", . . . , "*z*1". 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 *a*1 can authenticate CH_{a1}, and similarly for CH_{b1}, . . . , CH_{z1}. Fig. 3 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.

- ***Algorithm***

1. The sender of a message runs it through a MAC algorithm to produce a MAC data tag.
2. The message and the MAC tag are then sent to the receiver.
3. The receiver in turn runs the message portion of the transmission through the same MAC algorithm using the same key, producing a second MAC data tag.
4. The receiver then compares the first MAC tag received in the transmission to the second generated MAC tag.
5. If they are identical, the receiver can assume that the integrity of the message was not compromised, and the message was not altered or tampered with during transmission.

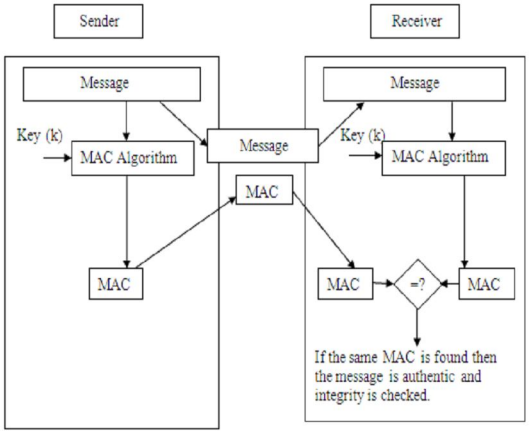

Fig. 4: MAC Generation

1. Source "s"

-Inter-cluster payload

P= Data| Mac (Data, $K_1^{inter, s}$)

MAC (Data, $K_2^{inter, s}$)… MAC (Data, $K_m^{inter, s}$)

-Node "s" forwards the inter-cluster packet to cluster heads

2. CH_a (cluster heads)

-Extract the MAC corresponding to its key

-Verify MAC (Data, $K_j^{inter, a}$)

-Packet $_a$ =Data |MAC (Data, $K_q^{inter, a}$) |$K_{q+1}^{intra, a}$| Header

-CH_a multicast Packet $_a$ to local receivers

3. Receiver "a" in the cluster of CH_a

-Wait for a packet from CH_a that contain $K_q^{intra, a}$

-Verify that $K_{q+1}^{intra, a}=H(K_q^{intra, a})$

-Verify MAC (Data, $K_q^{intra, a}$)

V. RESULTS

Fig 4: Source node displaying four destination nodes including original file content and its packet formation.

Fig 5: File browsed, Destinations 1 and 4 are selected to send packets .Content of file is shown in File Content textbox and 3 packets formed are displayed in Packet Formation text box.

Fig 6: Cluster Head1received the packets from source and forwarded to next cluster.

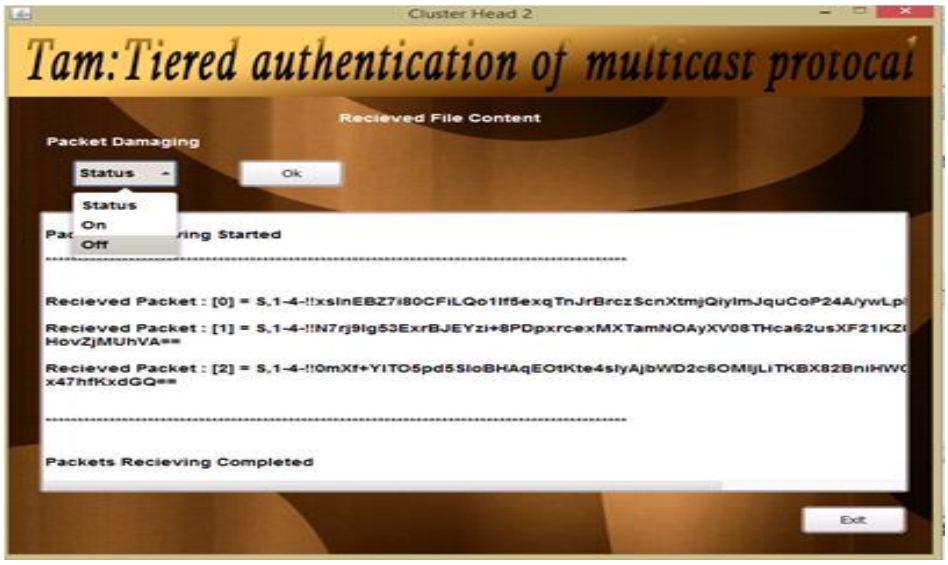

Fig 7:Clusted Head 2 received the packets .packet damaging status set to off and packets are forwarded to destinations

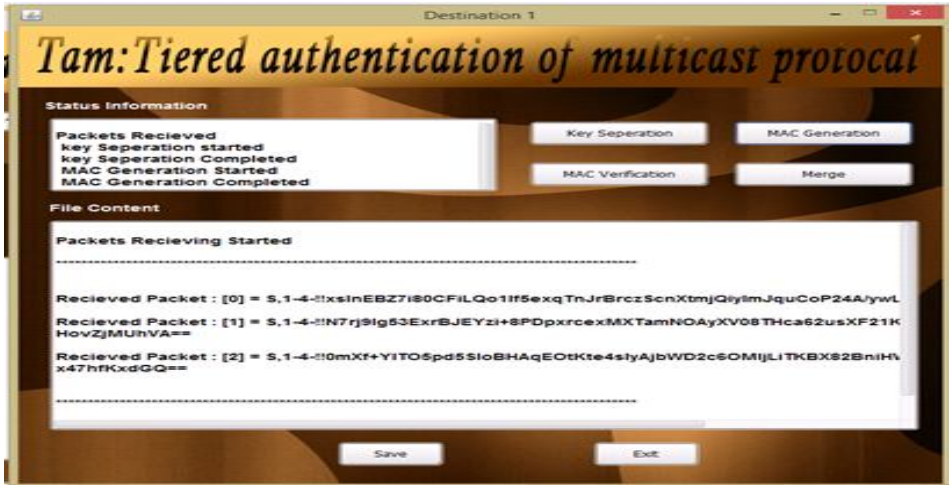

Fig 8: Destination 1 performs four task on the received packets i.e. Key Separation, MAC Generation, MAC Verification, and merging the packets to get the original file content.

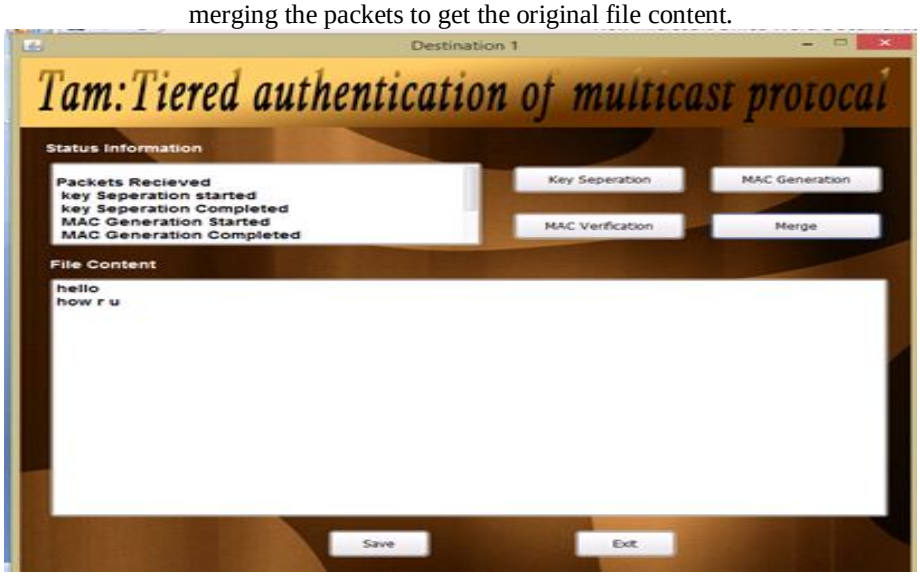

Fig 9: Destinations 1 successfully received the original file content.

ISSN: 2349-2163
www.ijirae.com

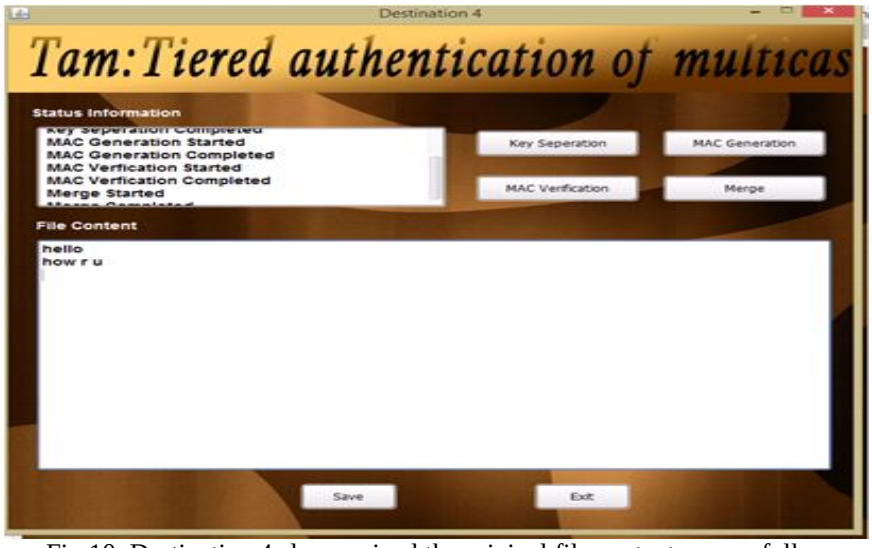

Fig 10: Destination 4 also received the original file content successfully.

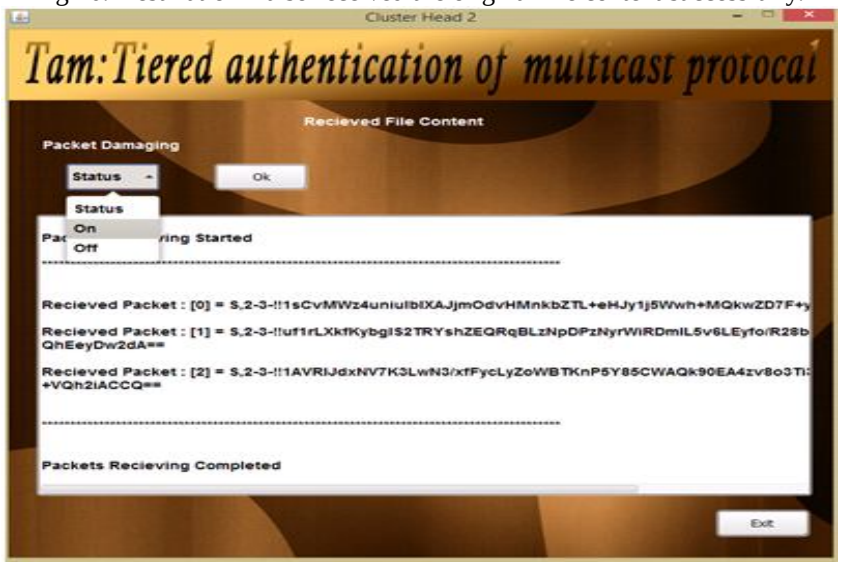

Fig 11: Here packet damaging status is set to ON to check the responses at the receiving destinations.

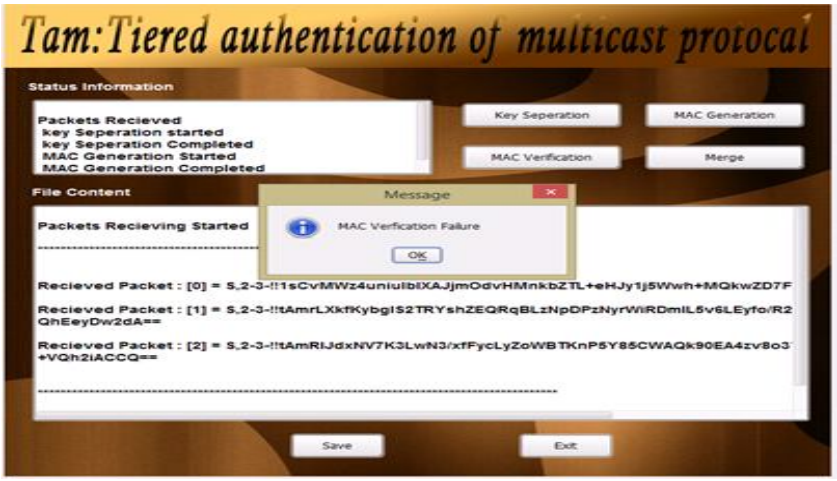

Fig 12: Destination received a damaged packets so results in MAC Failure and thus unsuccessful in receiving the original data.

VI. CONCLUSION AND FUTURE WORK

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 Etc. 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.

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.Proceedings of the 5th International Conference on Wireless and Optical Communications Networks, Surabaya, India, 2008; pp. 1–5.
[11]Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: a secure ondemand routing protocol for ad hoc networks. In Proceedings of the 8th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pages 12–23, September 2002.
[12] Yih-Chun Hu, Adrian Perrig, and David B Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In the Proceedings of Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, April 2003.
[13]D. Johnson, D. Maltz, and J. Broch. DSR The Dynamic Source Routing Protocol for Multichip Wireless Ad Hoc Networks, chapter 5, pages 139–172. Addison- Wesley, 2001.
[14]. Allirani, A.; Suganthi, M. An Energy Efficient Cluster Formation Protocol with Low Latency In Wireless Sensor Networks. World Acad. Sci., Eng. Tech. 2009, 51, 1–7.
[15].Muruganathan, S.; Ma, D.; Bhasin, R.; Fapojuwo, A. A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks. IEEE Radio Commun. 2005, 43, 8–1.

