Efficient Source Authentication for Multicasting in MANETs

John S. Baras and Prabha Ramachandran
Department of Electrical and Computer Engineering
and the Institute for Systems Research
University of Maryland, College Park, MD 20742, USA

[1] ***Abstract*— In this paper, we investigate a popular source authentication scheme, TESLA (Timed Efficient Stream Loss-tolerant Authentication) for multicast communication in mobile ad hoc networks. We evaluate the performance of a source authentication scheme inspired from TESLA and μTESLA based on simulations. Time synchronization is a crucial assumption made by these schemes. We describe effective means to achieve time synchronization in MANETs and conclude that overlay networks, whether they are UAVs or satellites are essential for high efficiency and performance. Since such overlay networks are typical in hierarchical wireless mobile networks, the proposed schemes are very appropriate for these scenarios. We claim that having an overlay network will help in routing and key distribution, in addition to time synchronization.**

I. Introduction

A mobile adhoc network (MANET) is a dynamically changing multi-hop network created by a set of mobile nodes that communicate either directly or indirectly via wireless links without relying on any centralized authority or fixed infrastructure[5], [6], [7]. A tactical MANET typically used for military applications in the battlefield scenario involves extensive group communication among the nodes and operates in a very hostile and demanding environment making it security-sensitive. Securing such MANETs is a Herculean task due to the vulnerability of the wireless links, poor physical protection of the nodes, dynamically changing topology and the absence of a fixed infrastructure apart from bandwidth and energy limitations. The security-related requirements of such MANETs have been discussed in [5], [7], [20]. In this paper, we concentrate on source authentication schemes for group communication. Several authentication mechanisms, varying from transmitting passwords in the clear to digital signatures based on secret or public key cryptography have been proposed and used in the past. For the unicast case, a simple keyed MAC can be used to check message integrity and verify sender authenticity. The problem is particularly hard in the multicast case since any receiver possessing the group key can forge packets. Solutions proposed for the multicast case include stream signatures, tree signing and multi-MAC methods [15], [16], [17], [18], [19]. In this paper, we evaluate the performance and discuss the applicability of one such source authentication scheme proposed by Perrig, *et al.* called TESLA [1], [2]. The rest of the paper is organized as follows. In Section II, we describe in brief TESLA and μTESLA and discuss effective means to achieve some implicit assumptions that the protocol makes. finalIn Section III, describe the simulation set-up and the results obtained. Section IV deals with security analysis. We conclude with Section V.

II. TESLA: Timed Efficient Stream Loss Tolerant Authentication

TESLA is computationally efficient as compared to signatures as it uses cryptographic primitives for authentication. It introduces the asymmetry through a delayed disclosure of keys. The scheme requires loose clock synchronization between nodes and relies on some form of authenticated exchange for bootstrap. TESLA has low computation overhead, low per-packet communication overhead, is tolerant to arbitrary packet loss, involves unidirectional data flow, needs no sender-side buffering and gives a high guarantee of authenticity and freshness of data [1], [2], [3].

The sender issues a signed commitment to a secret key K_i. The sender then uses that key to compute a MAC (Message Authentication Code) on a packet P_i and later discloses the key in packet P_{i+1}, which enables

[1] *Prepared through collaborative participation in the Collaborative Technology Alliance for Communications & Networks sponsored by the Army Research Laboratory under cooperative agreement DAAD19-01-2-0011.*

the receiver to verify the commitment and the MAC of packet P_i. If both verifications are successful, packet P_i is authenticated. The commitment is realized via a one-way, collision-resistant pseudorandom function(PRF). $K_i' = F(K_i')$ is the secret key used to compute the MAC of the next packet, and $F(K_i)$ commits to the key K_i without revealing it. The functions F and F' are two different pseudo-random functions. To bootstrap this scheme, the first packet needs to be authenticated with a regular digital signature scheme, *e.g.*, RSA. Packet P_{i+1} discloses K_i. The receiver first verifies K_i by checking if $F(K_i)$ and the commitment sent in packet P_{i-1} match. It also computes the MAC of the data in packet P_i using key, $K_i' = F'(K_i)$ to check the integrity of the data packet. Similarly, P_{i+1} is authenticated after the receipt of P_{i+2} [1], [2], [3].

Robustness to packet loss is achieved by using a one-way key chain rather than having the sender choose a key for each packet. To make it adaptive to dynamic sending rates, key is disclosed after d packets. In μTESLA, a single key is used to compute the MAC of all packets sent in a given interval [4]. Key K_i used to authenticate packets sent in interval i and is disclosed in interval $i+d$. Simultaneous use of multiple authentication chains with different disclosure periods helps accommodate heterogeneous receivers across the network [1], [2], [3]. TESLA is tailored for multicast. A new group member only needs to synchronize its time with the sender and receive the senders key disclosure schedule and a commitment to the key chain. An initial authenticated packet is still required to bootstrap the authentication process. TESLA uses a digital signature based periodic broadcast scheme for this purpose [1], [2], [3]. μTESLA [4] uses the node-to-Base-Station authenticated channel to bootstrap the authenticated broadcast.

A. *Time Synchronization*

In TESLA/μTESLA and other applications like SPINS[4] and Ariadne [9] whose authentication scheme is based on TESLA, loose synchronization among nodes is an implicit assumption made at the start. The term, ĺoosely time synchronizedḿeans that the synchronization does not need to be precise, but that the receiver must know a rough upper bound on the dispersion between its clock and the senderś. TESLA supports both direct and indirect synchronization. In direct synchronization, the receiver synchronizes its time directly with the data sender while in indirect synchronization, both the sender and the receiver synchronize their time with a common time synchronization service. Direct time synchronization involves message exchanges using nonces. It does not scale well as the sender becomes a bottleneck when there are a large number of receivers. There is also an added risk of a DOS attack at the sender.

For the MANET set-up, the time synchronization service must synchronize nodes without any message exchanges. The scheme must be reliable, robust to packet loss. Possible attacks include masquerade (spoofing time-server), tamper (modification of packet containing timing info), replay, DOS, delay. [11] Current day solutionsinclude terrestrial communication systems like T.V. and telephone (modems), direct radio broadcasts, navigation systems like GPS, Loran-C, Satellite Communication Systems like Two-Way Satellite Time Transfer (TWSTT) [11].

For most cases, a single satellite and can broadcast timing data to the entire MANET in a bandwidth and cost effective manner. The main disadvantage is the communication latency between two nodes connected by a satellite [11]. GPS provides timing accuracy in the 300ns range. Benefits include reliability, system-wide access, reduced calibration, installation and unit cost, small size and low power. However, the satellite transmission requires a line of sight between the receiver and the satellite. Any bounced signals, noise, nias and blunders can cause erroneous readings.

The cost of the receiving antenna at each MANET node can be reduced by having an intermediate (overlay) network between the satellite and the Manet-nodes. An overlay of Unmanned Aerial Vehicles (UAVs) is very common in tactical networks. The trusted high-power, high-memory UAVs can be used to synchronize time by indirect methods. This is analogous to the base stations in μTESLA. This scheme is thus scalable and suitable for multicast, enabling easy joining member time synchronization without any message exchanges between the new member and the sender(s). Since the UAVs are equipped with GPS, the line-of-sight and associated problems with noise are scaled down to a large extent. A single UAV can cover the entire MANET in most cases, else bidirectional overlay routing is also feasible. Each UAV has a common key, K for all the valid nodes of the MANET. The UAV broadcasts $time_{curr}, UAV_{ID}, MAC(K, time_{curr}|UAV_{ID})$ to its footprint. The MANET nodes can verify the MAC with the UAVś public key. The frequency of overlay broadcast should be adjusted so as to keep the time synchronization error between two receivers lesser than that tolerated by the authentication scheme.

B. *Bootstrapping*

Each receiver needs to be bootstrapped and given one authentic key of the one-way key chain as a commitment to the entire key chain and the key disclosure schedule of the sender. Methods for bootstrapping have been discussed in [2], [3], [4]. Since pre-loading bootstrap information is not approporiate for adhoc networks, secret key or signature based schemes should be used for bootstrapping. TESLA uses an expensive signature scheme for this purpose. While μTESLA avoids signatures and uses the node-to-basestation authenticated channel, it totally relies on the layer above for bootstrapping and involves message exchanges. Schemes involving exchanges between the sender and receiver are likely to have a bottleneck at the sender when many receivers try to bootstrap to the same sender and do not scale well. Ariadne [6] relies on a trusted Key Distribution Center(KDC).

If we deploy one or more high power nodes with two-way links to the overlay, the sender can send its bootstrap packet to the overlay, which would then send it to the members of the multicast group. Once the sender registers an initial bootstrap packet with the overlay network, new receivers can be bootstrapped easily by the overlay. Significant work has been done in [14] for hierarchical physical networks and for unidirectional routing. The receivers can wait for an authenticated broadcast from the overlay before contacting the sender for bootrsap information. Using the overlay for bootstrapping makes the scheme adaptive and reduces the bottleneck at the sender.

For authentication, the sender should use the secret key, K_{S-UAV} that it shares with the UAV to compute the MAC. The UAV can verify the MAC and be assured that only the sender could have generated the bootstrap packet, since K_{S-UAV} is known only to the sender and the UAV. The UAV then broadcasts this packet to the group members using a common key that it shares with all the MANET nodes to compute the MAC. This is somewhat similar to the base station authenticated broadcast in SPINS[4]. There are other added advantages of having an overlay network. The public-private key pair can be used to send information (like routing information or keys) to a specific node in the network. The UAV acts as a cluster-head with additional memory and storage capabilities. Overlay can be made to detect partitions and provide information to the nodes accordingly.

C. Sender Setup

Each sender pre-computes a sequence of secret keys (key chain) by choosing the last key K_n randomly and successively applying a one-way, collision resistant, strong cryptographic hash function, F. Sender associates each key of the key chain with one time interval and discloses the current key after a delay of $d = 2$ intervals after the end of the current time interval [4]. We use only one authentication chain for our simulations.

D. Receiver tasks

If a receiver is yet to be bootstrapped to a certain receiver, it waits for a UAV broadcast for a certain time and contacts the sender only when timeout occurs. When a node receives a bootstrap packet for a sender, it stores the packet in its buffer after verifying the MAC. The receiver computes the synchronization error from the information in the bootstrap packet. For every incoming packet, the receiver first verifies the security condition on receipt.

$$Interval_{ID}\left(t^R + Skew^{clk}_{S-R}\right) \leq (Interval_S + d) \quad (1)$$

Only packets that satisfy the security condition are buffered. For every key disclosure packet, the receiver verifies the security condition, checks the key authenticity using function, F, updates the key commitment the latest known TESLA key-interval id. It authenticates all packets sent between the interval idś of the last key disclosure packet and the current key disclosure packet after verifying the MAC. Keys for intermediate intervals can be computed using the latest key.

III. Security Analysis

The scheme guarantees source authentication and message integrity, i.e. the message could not have been modified in transit. The security condition takes care of an intermediate node turning malicious. Indirect time synchronization wards off Denial of Service (DOS) attacks at the sender. DOS at the receiver side can be created in many ways. Delayed packets will violate the security condition. Replay packets do not do much harm since a duplicated packet is accepted by the receiver only within a very short time period as the security condition is violated. Receivers reject packets if a malicious node tries to create a DOS attack by sending packets marked as being from an interval in the future as the security condition will be violated. Replay can be prevented by adding sequence numbers in the MAC [3]. However, the scheme cannot prevent a legitimate member from turning malicious and stop forwarding packets. It cannot detect a compromised node. It does not provide nonrepudiation. Neither does it prevent a node from generating a false route error message. It does not prevent all DOS attacks. Wormhole detection is also not an issue addressed by the authentication scheme. All these are issues that the routing protocol must handle.

Delay	12 nodes	22 nodes	32 nodes
avg	0.0334	0.0641	0.0959
max	0.0762	0.1507	0.2394

TABLE I

AVERAGE END TO END DELAY OF DATA PACKETS (MAODV)

Tint	12 nodes	22 nodes	32 nodes
0.25	0.5014	0.5042	0.5053
0.1	0.2514	0.2542	0.2849
0.05	0.2514	0.2727	0.2989
0.01	0.3024	0.2923	0.2815

TABLE II

AVERAGE BUFFER TIME PRIOR TO AUTHENTICATION WITH BUFFER TIME OUT SET TO 1.0 SEC

IV. SIMULATION RESULTS AND DISCUSSION

We used the Network Simulator 2, version ns-2.1b9a [12] to simulate TESLA. IEEE 802.11 was used as the Medium Access Control protocol. Routing was achieved using Multicast Adhoc Ondemand Distance Vector protocol(MAODV) [13], [8]. All nodes were assumed to be mobile with bidirectional connectivity. The channel capacity was 2 Mbps and the radio propagation range at the physical layer was set to $250m$. The nodes were placed in an area of $500m \times 500m$. Simulations were performed for 1200 seconds for three different group sizes (10, 20, 30) with 2 sources generating CBR packets at the rate of $4\ packets/sec$.

The metrics used to evaluate the performance of the authentication scheme were the percentage of packets received that are authenticated (dropped) and the delay due to buffering prior to authentication. Taking the MAODV routing delay $(TableI)$ into consideration, we set the dispersion to 0.001 seconds after studying the sensitivity by trial and error. The disclosure delay was set to 2 intervals and the buffer timeout was 1.0 second. Keeping these variables fixed, the TESLA time interval duration, T_{int} was varied to study the percentage of incoming packets that satisfy the security condition (% Buff), the percentage of buffered packets that are dropped due to buffer timeout $($% Buffdrops$)$, the percentage of packets received that are authenticated $($% Auth$)$ and the percentage of packets received that are dropped $($% Totaldrops$)$. Packets get dropped either due to violation of the security condition 1 or due to buffer timeout.

metric	12 nodes	22 nodes	32 nodes
%Buff	100	100	100
%auth	99.954	99.953	99.953
%buffdrops	0.0	0.003	0.016
%totaldrops	0.0	0.003	0.016

TABLE III

PERFORMANCE EVALUATION FOR $T_{int} = 0.25$ SECONDS

metric	12 nodes	22 nodes	32 nodes
%Buff	100	100	89.359
%auth	99.97	99.976	89.336
%buffdrops	0.0	0.0	0.118
%totaldrops	0.0	0.0	10.745

TABLE IV

PERFORMANCE EVALUATION FOR $T_{int} = 0.1$ SECONDS

metric	12 nodes	22 nodes	32 nodes
%Buff	100	78.837	51.869
%auth	99.97	78.813	51.847
%buffdrops	0.0	1.63	4.7
%totaldrops	0.0	22.45	50.56

TABLE V

PERFORMANCE EVALUATION FOR $T_{int} = 0.05$ SECONDS

metric	12 nodes	22 nodes	32 nodes
%Buff	29.03	14.35	8.828
%auth	29.02	14.349	8.824
%buffdrops	3.947	6.07	1.13
%totaldrops	72.11	86.51	91.27

TABLE VI

PERFORMANCE EVALUATION FOR $T_{int} = 0.01$ SECONDS

As expected, the routing delay increases linearly with group size $(TableI)$. Table II shows the average time that a packet spends in the buffer at the receiver before it is delivered to the application. This metric is computed as the percentage of buffered packets that get authenticated *i.e.*, packets that satisfy security copndition on arrival at receiver and whose key is disclosed in a subsequent packet that satisfies the security condition before the buffer times out. The buffer time increases linearly with group size for interval sizes of 0.25, 0.1 and 0.05. This is because the security condition is satisfied for most packets and hence the buffer time is only due to the delay in packet (key) arrivals owing to routing delay. Tables III, IV, VI, V show these packet counts in terms of the percentages described above for T_{int} duration of 0.25 seconds, 0.1 seconds, 0.01 seconds and 0.05 seconds respectively. For a time interval of 0.01 seconds, for all group sizes, % Buff is low. This is because of the fact that the routing delay is more than $T_{int} \times d$. Most buffered packets are authenticated since % Auth and % Buff are comparable for all cases. We also note that the number of bufer time out drops is less for al cases and hence a time out value of 1.0 seconds is very reasonable.

V. Conclusion and Future Work

[2] In this paper, we proposed TESLA and μTESLA based source authentication schemes as candidate multicast source authentication schemes for mobile adhoc networks. We evaluated TESLA and compared the authentication delay and percentage packet drops for differentscenarios. Except for the case when $T_{int} = 0.01$ seconds, TESLA performs reasonably well for all three group settings. The reason for poor values for 0.01 is that the product, $T_{int} \times d$ is less than end to end MAODV delay. Obviously, the security condition will be violated for a large fraction of incoming packets. TESLA performs well as long as therouting delays are reasonable. The performance of TESLA depends only on the delay in the multicast routing scheme used. Thus, we conclude that TESLA is suitable for multicast settings in adhoc networks. As part of our future work, we hope to demonstrate the performance gain achieved in MANETs in the presence of overlay for bootstrapping and time synchronization. We also plan to simulate TESLA over ODMRP and compare the performance with MAODV. We intend to make all measurements for two set-ups: a flat MANET and one with overlay; compare and contrast the above metrics.

VI. Acknowledgements

The authors would like to thank Huigang Chen, Harsh Mehta, Ayan Roy-Chowdhury, Nalini Bharatula and Lakshmi Ramachandran for their support in making this work possible.

References

[1] A. Perrig, R. Canetti, B. Briscoe, J.D. Tygar and D.Song, “TESLA: Multicast Source Authentication Transform”,*Internet Draft, Internet Engineering Task Force*, draft-irtf-smug-tesla-00.txt, November 2000.

[2] A. Perrig, R. Canetti, J.D.Tygar, and D.Song, ”Efficient authentication and signing of multicast streams over lossy channels”, *In IEEE Symposium on Security and Privacy, May 2000*, pp. 56-73.

[3] A.Perrig, R.Canetti, D.Song, J.D.Tygar,” Efficient and Secure Source Authentication for Multicast”,*In Network and Distributed System Security symposium, NDSS 0́1*,Feb 2001.

[4] A.Perrig, R.Szewczyk, V.Wen, D.Culler, J.D.Tygar,”SPINS: Security Protocols for Sensor Networks”, *In proceedings of MobiCom 2001*, Rome, Italy, July 2001.

[5] L. Zhou and Z.J.Haas, ”Securing Ad Hoc Networks”, *In IEEE Network magazine, special issue on networking security*, vol 13, No. 6, pp. 24-30,November/December 1999.

[6] Jean-Pierre Hubaux, L.Buttyan and S. Capkun,” The Quest for Security in Mobile Ad Hoc Networks”,*In the proceedings of the ACM symposium on Mobile Adhoc Networking and Computing (MOBIHOC),2001*

[7] S. Jacobs and M.S. Corson,”MANET authentication architecture”,*Internet Draft*, August 1998.

[8] M. Royer and C.E. Perkins, “Multicast operation of the adhoc on-demand distance vector Routing Protocol”,*In the Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM),*pages 201-218, 1999.

[9] Y-C Hu, A.Perrig, and D.B. Johnson,” Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks”,*Technical Report TR01-383*, Department of Computer Science, Rice University, December 2001http:www.monarch.cs.rice.edumonarch-papersariadne.ps

[10] M. Bishop,” A security analysis of the NTP protocol”,*www.eecis.udel.edu/ ntp/ntp_spool/doc/security.ps.gz*

[11] Y. Zhang, D. DeLucia, B. Ryu and S. Dao, ”Satellite Communications in the Global Internet: Issues, Pitfalls, and Potential”,*INET’97*, Kuala Lumpur, Malaysia, June 1997.http://www.wins.hrl.com/people/ygz/papers/inet97.html

[12] NS2 Manual and Documentation,*http://www.isi.edu/nsnam/ns/ns-documentation.html.*

[13] NS-2 MAODV Simulation code: *http://www4.cs.uni-dortmund.de/ Lindemann/*

[14] Young-bae Ko and N. H. Vaidya,“A Routing Protocol for Physically Hierarchical Ad Hoc Networks”,*Technical Report 97-010*, Computer Science, Texas A&M Univ., September 1997.

[15] R.Canetti, J.Garay, G.Itkis, D.Micciancio, M.Naor and B.Pinkas, “Multicast security: a taxonomy and some efficient constructions*IEEE INFOCOM 9́9*Mar. 1999, pp.708-716.

[16] D.Boneh, G.Durfee and M.Franklin,”Lower bounds for multicast message authentication,”*Eurocrypt 2001*, May 2001, pp.437-452

[17] R.Gennaro and P.Rohatgi,”How to sign digital-streams”,*Advances in Cryptology (CRYPTO 9́7)*,Aug. 1997, pp.180-197.

[18] R.Merkle, “A certified digital signature:”,*Advances in Cryptology (CRYPTO 8́9)*,Aug. 1989, pp. 218-238

[19] P.Rohatgi,”A compact and fast hybrid signature scheme for multicast packet authentication,”6th *ACm Conference on Computer and COmmunications Security,*”Nov. 1999, pp.93-100

[20] T.Hardjono and G. Tsudik, “ IP Multicast SEcurity: Issues and Diretcions”,*Annales de Telecom, 2000.*

[2] *The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or employed, of the Army Research Laboratory or the U.S. Government.*