CACMAN COMPARISION WITH MOCA USING PKI ON MANET.

[1] MR. NIRAV N. KUBAVAT,[2] PROF. S.M.MANIAR

[1]M.E. [Computer Engg] Student, Department Of Computer Engineering, V.V.P. Engineering College, Rajkot, Gujarat
[2] Asst.Professor, Department Of Computer Engineering, V.V.P. Engineering College, Rajkot, Gujarat

neeravkubavat@gmail.com , sweetymaniar@yahoo.com

ABSTRACT: *MANET applications and services pose many interesting challenges due to their unique features. Specifically, security is getting a lot of attention in every aspect of MANETs due to their inherent vulnerability to attacks. Threats exist in every layer of the MANET stack, and different solutions have been adapted for each security problem. Another problem for MANETs is availability, and adding more resources will not necessarily make the system more available. Certificate Authority (CA) is one of the most important entities in Public Key Infrastructure (PKI) and needs to be designed carefully when adapted to MANETs. The main goal of our work is to provide a framework that addresses the issues of performance and security of CA in MANETs. Additionally, we would like to increase the availability of CA services, while lowering packet overhead of the network, without increasing the network vulnerability.*
In this paper, we present a framework suitable for exchanging PKI certificates in MANETs. By caching and exchanging certificates between clients collaboratively, we will show that our system can meet the performance challenges of providing CA service without sacrificing system security.

Keywords— PKI, Key Management, Security, MANET, Ad Hoc Networks, Threshold Cryptography.

I: INTRODUCTION

MANETs are raising many interesting challenges for applications and services due to their unique characteristics. Not relying on fixed infrastructure combined with tight constraints, such as power consumption, transmission range, and possibly computational capabilities are examples of such challenges. On the other hand, security is as crucial as other challenges and has received much attention from researchers in every aspect of MANETs due to their inherent vulnerability to wider security attacks compared to wired networks.

PKI is an important service that provides a security framework to the system using certificates and public key encryption/decryption techniques. Typically, a PKI system consists of certificates, CAs, a revocation mechanism, and a mechanism to evaluate a chain of certificates to the target [5]. Certificates are used for encrypting or signing in many vital applications, such as authentication, exchange of routing info, encrypting or signing email, and much more. A CA is usually used to organize, store, and issue those certificates. Adopting PKI in MANETs is not an easy task, since PKI is mostly designed for centralized, wired, and well-connected networks. The introduction of MANETs makes the task of providing a reliable and secure service much more difficult.

Recently, researchers have identified these constraints and tried to provide some solutions to adapt CA for MANETs (e.g. [1],[6], and [7]). These solutions rely on secret sharing mechanisms [8][9][10] to increase security and availability, since installing the CA service in just one node will make it vulnerable and exact replication of the CA will make the situation even worse[6]. Despite the fact that secret sharing seems to be a natural fit for MANET, it cannot be deployed without consequences that will be discussed shortly.

The aim of this work is to minimize the burden of adapting CA services in MANETs by minimizing the packet overhead and maintaining high availability of the service at the same time. This goal has been achieved by allowing clients to share some responsibilities with CA servers by cooperatively caching a portion of the certificates generated by CA servers. The characteristics of certificates can make caching a reasonable solution to the availability problem. Our cashing-based framework will address security and performance challenges of providing CA services in MANETs. In addition, it suggests techniques with minimal overhead that can help enhance our main goal, availability, without compromising the security of the network. We will show the feasibility of our framework and compare it to related work that address the same problem.

II: Related Work

Mobile CA (MOCA)[6] has been introduced as the matching part of the Cornell Online Certificate Authority (COCA) [12] framework, but with the consideration of MANETs' unique properties, since the latter was originally designed for wired networks and did not consider the connectivity of clients. MOCA is a mobile node within an ad-hoc network selected to provide distributed CA functionality. MOCA nodes apply threshold cryptography to share the responsibility and increase availability of the ad-hoc network. When a client wants to obtain a certificate, it sends Certificate Request (CREQ) packets to at least t MOCA servers and waits for a reply. Any MOCA that receives a CREQ will reply by sending a Certification Reply (CREP) packet containing its partial signature. Once the client receives t valid CREPs, it can reconstruct the whole certificate. CREQ and CREP have been embedded in Route Request (RREQ) and Route Reply (RREP) messages that are found in on-demand ad-hoc routing protocols like AODV [13] and DSR [14] to reduce the amount of overhead packets. MOCA increases availability by letting clients send β unicast CREQ, where $\beta = t + \alpha$ and α is determined in each client by inspecting the status of the network. Additionally, to avoid flooding the network, MOCA clients inspect the route table to see if any cached routes to β MOCAs are available, making flooding a last resort when the cached routes are less than β . On the other hand, Kong et al.'s [15] goal is to provide pervasive CA services by making all n nodes in the network share CAs' functionality. However, we will not compare the work due to the same reasons mentioned in [6].

Schemes Metrics	Flooding	unicast
Packet Overhead	High	Low
Response Time	Long	Quick
Success ratio	High	Medium
No. of CREP Received	Greater	Medium
High CA Threshold Value	Best	Worst
Low CA Threshold Value	Average	Best
Medium CA Threshold Valure	Average	Average

Table 1 : Comparison of Flooding and Unicast for CA

III: CACMAN

We showed in [7] two concerns about MOCA, which we believe may reduce the usability of the framework. The first concern was that the number of MOCA nodes is relatively high, about 10 to 20% of the total number of participants. We believe that making n that large does not come without some consequences (e.g. overhead of CA synchronization and key refreshing). The second concern was that the ratio of control packets generated by MOCA, the sum of CREQ and CREP, compared to flooding is high. In the best case, MOCA saved about 30% of packet overhead when 5=ß and only about 5% when 25=ß (ß=t and 30=n) [6]. The main idea of CACMAN is to make clients play a more active role in CA services by giving them some responsibilities, namely, caching valid certificates for their usage and giving them to other clients when necessary. This will reduce the burden on CAs and reduce the need of adding more replicas to increase availability, since doing that, with fixed threshold, will make the system more exposed and its consistencydifficult to maintain. In addition, CACMAN increases the availability and efficiency of the system without any additional CA servers by caching combined and partially signed certificates in each client's local memory. CACMAN still needs CA servers to generate new certificates and revoke them. However, their number, when CACMAN is used, will not play a significant role for in system availability.

The NS-2 simulator gives us more insight into the problem from the performance prospective. Here, we will show the basic revised CACMAN model, which differs slightly from the one introduced in [7].

When a client wants to obtain other client(s)' certificates to establish a secure communication or to encrypt some messages, it will perform the following steps:

1- It checks its own cache to find out the availability of a complete certificate, or it will identify the missing parts if partially signed certificates are found.

2- If there is no sufficient number of partially signed certificates, then it makes a local broadcast. However, if it happens that the source has a short route to the subject in question; it should favor sending the CREQ to that subject instead of making a local broadcast[1].

3- Every client that receives a CREQ message inspects its cache for a possible hit. It sends a CREP back to the sender if a full or partial certificate(s) is found for the subject. We have also tried another variation for this step named 'CACMAN-X replies'. The difference is that if no full certificate has been found, then X randomly selected shares will be sent. The goal of this is to increase the chance of delivering more desired partials. In this paper, we tried X=2, 4, 8, and All.

4- The request initiator waits for a specified time; if it receives sufficient responses then it reconstructs the certificate. Notice that the client could receive a complete certificate, which eliminates the reconstruction step but not the verification. If the request initiator times out because insufficient partials have been received or the reconstruction has failed, then the client may try to contact CA servers by flooding the network or by sending unicast messages if it has sufficient routes to them. Otherwise, it reports a failure to the client or retries later.

Schemes Metrics	CACMAN	MOCA
Packet Overhead	Low	High
Response Time	Quick	Long
Success ratio	High	Medium
No. of CREP Received	High	Medium
Flooding Ratio	Very Low	Very High
Certificate Availability	Easy	Difficult
Security	High	Low
Caching Certificate	Yes	No

Table 2 : Comparison of CACMAN and MOCA

IV: Simulation and Discussion

In our simulations, we have used the most recent build of NS-2 to simulate the following hypothetical scenario similar to the one used in [6]: 150 mobile nodes scattered in a 1000 m^2 field, 30 of which are CA servers[2] to the remaining 120 clients' nodes. All nodes are moving using the CMU random waypoint model[16]. One hundred nodes will make 10 CREQs during the 600 second simulation time. We have generated nodes mobility scenarios for maximum speeds of 0, 1, 5, 10, and 20 m/sec with pause times of 0 and 10 seconds. Each configuration is replicated at least three times. The reason for choosing this configuration is to compare our work with the MOCA framework. In addition, we have used four cache sizes- 75, 150, 225, and 300 share slots - and four thresholds for key construction - 5, 15, 20, and 25. For the purpose of comparing our work with MOCA, their figures are shown in Figure 1 and Figure 4(a), (b).

Total Number of Mobile Nodes	150
Number of MOCAs	30
Area of Network	1000m x 1000m
Total Simulation Time	600 seconds
Number of Certification Requests	10 requests each from 100 non-MOCAs
Node Pause Time	0, 10 seconds
Node Max. Speed	0, 1, 5, 10, 20 ms
Cache Sizes	75,150,225,300

Table 3: Simulation Parameters

Figure 1(a) shows the number of received CREPs for pure flooding certification protocol. It is effective; however, it injects a large number of control packets into the network. Figure 1(b) shows the effect of introducing the MOCA protocol. The peaks around different betas are caused by MOCA when it uses cached route information to MOCA servers. As we can see, the occurrences are still shifted to the right, which indicates a high rate of success, but many redundant replies are also received.

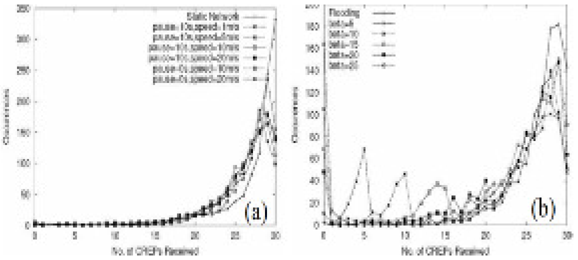

Figure 1. (a)Flooding based certification protocol
(b) MOCA certification protocol

On the other hand, Figure 2 and Figure 3 show the basic CACMAN protocol. Each figure shows the effect of mobility and cache sizes when we fix the threshold. The legends in these figures are interpreted as 'number of nodes-pause time-max speed (cache capacity) protocol'. In Figure 2, when T=5, very few requests did not get sufficient replies, and the majority received an overwhelming number of replies. When we increased the threshold to 15, as in Figure 3, the number of requests that did not get sufficient replies increased and the cache size played a more important role.

Notice that when the cache capacity increases to 150 and 225, as in Figure 2 (b and c)respectively, the situation is much better as the number of unsuccessful requests decreases.

Due to lack of space, we will consider only the case where T=15 with max speed of 10 m/s and 0 pause time as a typical case.

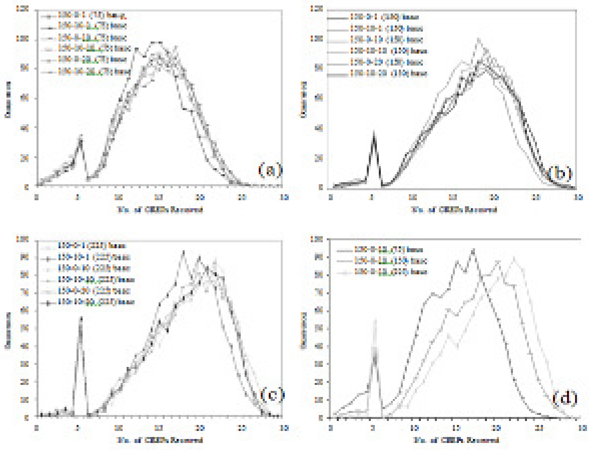

Figure 2.(a-c) Mobility effect with different cache sizes.(d)Cache capacity effect (T=5).

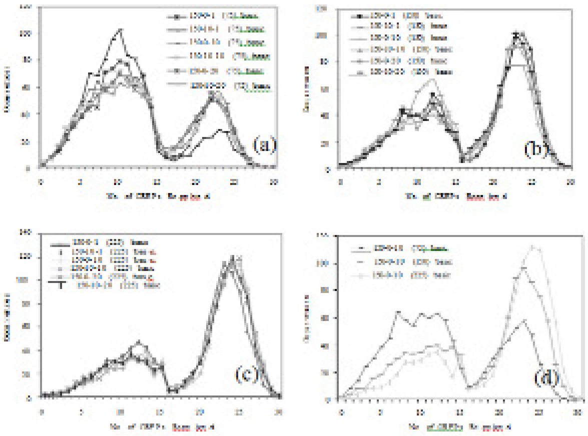

Figure 3.(a-c) Mobility effect with different cache sizes.(d)Cache capacity effect (T=15).

From the above figures we can observe the following points for CACMAN.

1- The effect Of the threshold is noticeable, and the more it increases, the less the likelihood of getting all partials from the first attempt.

2- The peaks around the thresholds are due to the following reasons:

   a. The requester received or already has a complete certificate (not partial), so it will be counted as if it has received or had T shares.

   b. The requester has, before sending the CREQ, some partials in its cache but is missing few. When it sends a CREQ, many of the replies are already cached, but the missed parts eventually arrive, and this will be reflected as a peak around the threshold since we do not count duplicate partials.

3. The mobility effect is not that significant since CACMAN utilizes local broadcasts, and mobility will have a significant effect when routing information is involved in the protocol.

4- Cache size obviously has an impact, as shown in (d) of Figure 2 and Figure 3. However, going from 75 cache size to 150 is more significant and effective than going from 150 to 225, and this is consistent in other thresholds and max speeds, which suggests 150 as the best deal for the cache size and hit ratio tradeoffs.

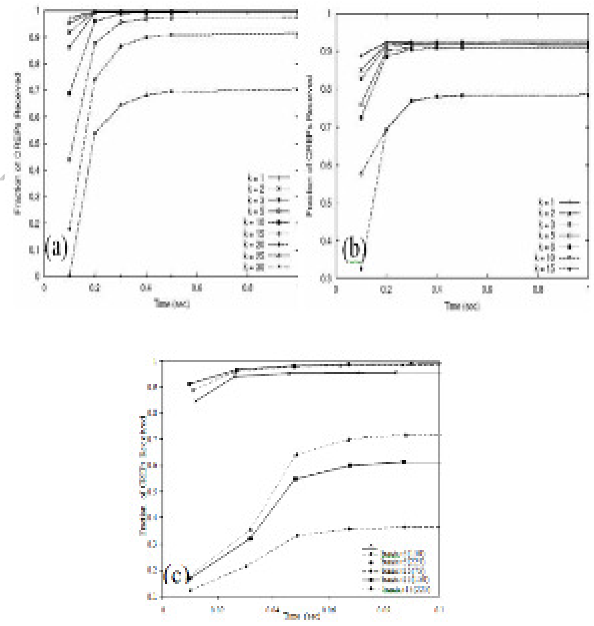

Figure 4. Success Ratio with (a) Flooding (b) MOCA Closest-Unicast (c) CACMAN basic.

Figure 4 shows the success ratio for various scenarios[4]. We used the same technique used in [6] to plot the CACMAN success ratio. By observing this figure, flooding is indeed the most effective way for obtaining a high success ratio, but with the price of packet overhead. CACMAN and MOCA perform similarly to each other for threshold 5 and 15. The cache size has a noticeable effect when T=15, and the use of X-replies optimization tightens the gap between cache sizes. By observing the time scale, CACMAN stabilizes much faster than flooding and MOCA due to the fact that some partials are available at the local cache of the requester.

Besides, CACMAN requests never propagate more than one hop.

V: Conclusions

In this paper, we have presented CACMAN, a framework for enhancing CA service in MANETs, and shown how cooperative certificate caching would play a pivotal role in decreasing the overhead of offering certification service, relieving CA servers, and still maintaining high availability. In addition, we showed one effective stateless way of accessing certificates using local broadcasting.

We are interested in finding some applications other than PKI that could benefit from the introduction of client caching in a one-to-many-to-one communication paradigm. Currently, we are fine-tuning enhancements suggested in section 5 in our simulation environment in order to see their effect on the performance of CACMAN. A future direction is to investigate other types of networks, such as hybrid and wired Internet peer-to-peer networks, and see how effective it is to deploy CACMAN in these environments.

References

[1] L. Zhou and Z.J. Haas, "Securing Ad-hoc Networks",IEEE Network Magazine, Nov. 1999.

[2] L. Bononi et al., "A Differentiated Distributed Coordination Function MAC Protocol for Cluster-Based Wireless Ad-Hoc Networks", Proc. 1st ACM int'l Workshop on Performance Evaluation of Wireless Ad-Hoc, Sensor, and Ubiquitous Networks, Venezia, Italy, Oct. 2004, pp. 77 - 86.

[3] M.C. Domingo and D. Remondo, "An Interaction Model between Ad-Hoc Networks and Fixed IP Networks for QoS Support", Proc. 7th ACM int'l Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Venice, Italy, Oct. 2004, pp. 188 – 194.

[4] W.H.O. Lau, M. Kumar, and S. Venkatesh, "Mobile Ad Hoc Networks: A Cooperative Cache Architecture in Support of Caching Multimedia Objects in MANETs", Proc. 5th ACM int'l Workshop on Wireless Mobile Multimedia, Sept.2002.

[5] C. Kaufman et al., Network Security: Private Communication in a Public World, Prentice Hall, 2nd ed.,2002.

[6] S. Yi and R. Kravets, "MOCA: Mobile Certificate Authority for Wireless Ad-hoc Networks", 2nd Annual PKI Research Workshop (PKI03), Apr. 2003.

[7] L. Al-Sulaiman and H. Abdel-Wahab, "Cooperative Caching Techniques for Increasing the Availability of MANET Certificate Authority Services", The 3rd ACS/IEEE Int'l Conf. Computer Systems and Applications (AICCSA- 05), Cairo, Egypt, Jan. 2005.

[8] Y. Desmedt, "Society and Group Oriented Cryptography: A New Concept", In C. Pomerance, ed., Advances in Cryptology Crypto '87 Proceedings, no. 293, LNCS, Santa Barbara, CA, Springer-Verlag, 1988, pp. 120-127.

[9] S. Jarecki, Proactive Secret Sharing and Public Key Cryptosystems, Master Thesis, MIT, 1995.

[10] Shamir, "How to Share a Secret", Communications of the ACM, vol. 22, no. 11, Nov. 1979, pp. 612-613.

[11] S. McCanne and S. Floyd, The LBNL Network Simulator (NS-2); http://www.isi.edu/nsnam/ns/.

[12] L. Zhou, F. Schneider, and R. van Renesse. "COCA: A Secure Distributed On-line Certification Authority", ACM Trans. on Computer Systems, vol. 20, no. 4, Nov. 2002, pp.329-368.

[13] E. Perkins and E.M. Royer, "Ad-Hoc On-Demand Distance Vector Routing", In The 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, Feb. 1999, pp. 90-100.

[14] J. Broch and D.B. Johnson, "The Dynamic Source Routing Protocol for Mobile Ad-Hoc Networks", IETF Internet Draft, Feb. 2003.

[15] J. Kong et al., "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks", In Proc. ICNP '01, 2001, pp. 251-260.

[16] J. Broch et al., "A Performance Comparison of Multi- Hop Wireless Ad-Hoc Network Routing Protocols", Proc. 4th annual ACM/IEEE int'l conf. Mobile Computing and Networking, Dallas, TX, Oct. 1998, pp. 85 – 97.

[17] E. Pagani and G.P. Rossi, "A Framework for the Admission Control of QoS Multicast Traffic in Mobile Ad- Hoc Networks", Proc. 4th ACM int'l workshop on Wireless Mobile Multimedia, July 2001, pp. 2-11.