BARTER: Profile Model Exchange for Behavior-Based Access Control and Communication Security in MANETs

Vanessa Frias-Martinez, Salvatore J. Stolfo, and Angelos D. Keromytis

Computer Science Department, Columbia University
500 West 120th Street, New York, NY 10027
{vf2001,sal,angelos}@cs.columbia.edu *

Abstract. There is a considerable body of literature and technology that provides access control and security of communication for Mobile Ad-hoc Networks (MANETs) based on cryptographic authentication technologies and protocols. We introduce a new method of granting access and securing communication in a MANET environment to augment, not replace, existing techniques. Previous approaches grant access to the MANET, or to its services, merely by means of an authenticated identity or a qualified role. We present BARTER, a framework that, in addition, requires nodes to exchange a model of their behavior to grant access to the MANET and to assess the legitimacy of their subsequent communication. This framework forces the nodes not only to say who or what they are, but also how they behave. BARTER will continuously run membership acceptance and update protocols to give access to and accept traffic only from nodes whose behavior model is considered "normal" according to the behavior model of the nodes in the MANET. We implement and experimentally evaluate the merger between BARTER and other cryptographic technologies and show that BARTER can implement a fully distributed automatic access control and update with small cryptographic costs. Although the methods proposed involve the use of content-based anomaly detection models, the generic infrastructure implementing the methodology may utilize any behavior model. Even though the experiments are implemented for MANETs, the idea of model exchange for access control can be applied to any type of network.

Keywords. IDS in Mobile Ad-hoc Networks, IDS cooperation, Anomaly Detection, Access Control, Threshold Cryptography

1 Introduction

We propose a novel approach to secure nodes in Mobile Ad-hoc Networks (MANETs) based on the use of anomaly detection sensors. We put forward that each node

* This work has been partially supported by a grant with NIST; the Army Research Office Contract #DA W911NF0410442; and the DARPA Grant HR0011-06-1-0034. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

in a network computes a profile of its own behavior and provides this behavior model to request and gain access to services. A device will not only authenticate a node on the basis of standard techniques, but will also compare the model against normal expected behavior before granting access.

MANETs differ from wired networks in that there is no central control, no base station, and no wireless switches, with devices entering and leaving the network dynamically, quickly changing the network topology and administrative domain membership. Given that, it seems clear that it is important to provide services like authentication [20], access control [12], and key exchange [5], among others. We study two important issues in MANETs: access control and update control. Access control refers to the initial acceptance of devices to the MANET, and update control to the membership update of the devices that are already part of the MANET.

Previous work in access control and update control lies mostly in the realm of Role-Based Access Control (RBAC) [6]. Role-based access control executes the acceptance and update control based on the roles that each user/device has within the organization. More complex RBAC implementations define policies that may be modified over time, but always via human intervention. For example, Lin *et al.* [14] proposed an RBAC system where initial policies implemented by authorities may be changed via trust policy description files (TPDFs), but these changes need to be made manually. In order to secure trust-sensitive operations and validate delegations, many RBAC approaches have introduced a PKI infrastructure [7]. Although RBAC offers a secure access control infrastructure for organizations with fixed roles, it does not scale well to dynamic environments (MANETs) where behaviors and policies may have to change often and where changes cannot be predicted over time, making it very difficult for humans to manually create policies online.

Most of the work described for RBAC focuses on wired networks. At a wireless level, and at an ad-hoc wireless level, the vast majority of work implemented for access control and update focuses on the implementation of threshold cryptographic approaches. Threshold cryptography distributes the development of certificates and PKIs among t devices in a MANET. Both certificates and shared keys (distributed PKI) allow the MANET to control, in a distributed manner, the acceptance of new members via collaborative creation of certificates (to identify that the user local keys are really their keys) and collaborative creation of distributed secret keys (to secure the communication channel among devices of the MANET). Any t random devices in the MANET will be able to recover these secrets via Lagrange interpolation, when needed. Distributed certificates can also be revoked at any moment by any device in the MANET, or simply not renewed after a certain time. Distributed shared secret keys are periodically regenerated (proactive shared key [9]) to avoid attackers that may be eavesdropping the channel for long periods of time. These works do not typically specify how the acceptance or rejection decision is made. Kong *et al.* [11] proposed a threshold cryptographic system for MANETs that supports ubiquitous security services and scales as the size of the networks increases. This work is mostly

based on Herzberg *et al.* [9] who explored proactive secret sharing and key regeneration, and on Pedersen [20] that tackled key regeneration with no trusted party. Although threshold cryptography is necessary to secure MANETs, such systems also need an infrastructure to decide on the acceptance or rejection of devices to the MANET.

We present BARTER, a behavior-based access control and communication security framework for MANETs that automatically updates the acceptance and rejection policy based on the real-time behavior changes of its members modeled by anomaly detection sensors. BARTER enhances MANET access control schemes by allowing nodes to automatically decide how to change their membership policies without human intervention. BARTER is built on top of a threshold cryptographic infrastructure that guarantees fully distributed decision making and secure communication among peers. The BARTER framework consists of three phases: during *bootstrap*, all the devices that start the MANET agree on a certain common normal behavior; second, during the *membership acceptance*, by exchanging models among nodes, a node not only announces its identity, but also provides a description of its typical behavior. This behavior is used by other nodes to grant or deny access to the MANET. Third, during the *membership update* (communication phase) the models exchanged are also used to detect anomalous content in the traffic shared between any two nodes in the MANET. We present experimental results with ENRON [4] email profiles that successfully merge a threshold cryptographic approach with the BARTER framework. In our experiments, we analyze the tradeoff between the threshold cryptographic value t (fixed or variable) and BARTER's performance during the access control. The BARTER framework reaches –during access control– false rejection rates of 0.03% with true rejection rates above 90% and cryptographic costs around 0.24 seconds per device accepted.

In the following sections, we provide a detailed view of how one may define a profile (behavior model), how these may be compared to grant access, and how they may be used to test traffic. Some of our preliminary results have been presented in Cretu *et al.* [3]. We are not aware of previous work that aims to use *model exchange* as a security feature. The closest concept was developed by Necula and Lee [17][18] in their Proof-Carrying Code (PCC) approach, where the producer (client), was required to create a formal safety proof of its own code to conform with the safety policy defined by the server (consumer). However, our approach differs in the fact that devices exchange behavior models and not safety proofs. Also, a similar idea was recently introduced by the IEEE 1667(TM) standard [10] to authenticate USB devices before downloading information from them to a local machine. Unlike BARTER, that work focuses on USB devices, without the complexities of a wireless ad-hoc environment.

The rest of this paper is organized as follows: in Section 2 we describe the main features of the BARTER framework (threat scenarios and anomaly sensor that we use for our experiments). In Sections 3, 4 and 5 we describe the three phases of BARTER. Section 6 offers some experimental results using this framework, and Section 7 presents the conclusions and future work.

2 BARTER Framework

BARTER implements a behavior-based access control and communication security framework for mobile ad-hoc networks. BARTER works under the assumption that each device in the MANET is running an Anomaly Detection sensor that allows the device to model its local input and output behavior. The main goal of BARTER is to enhance existing access control and communication security layers by means of exchanging behavior models that are used to accept new devices to the MANET and to check the security of subsequent communications.

BARTER consists of three main phases: *bootstrap*, *membership acceptance* and *membership update*. The *bootstrap* phase is executed only once, during the creation of the MANET. We assume that each device's input and output profiles represent an initial model of what normal behavior is for the MANET (although it may change over time). During bootstrap, all initial members exchange their profiles and compute the decision of what locally normal is for each of them and for the group. This group of devices is initially responsible for admitting new devices to the MANET. The *membership acceptance* phase takes place when a new node approaches a MANET and requests access to its services. The protocol executed during this phase tests whether the profile of the new, requesting node is similar to the profiles of the nodes that are already part of the MANET, and that have exhibited good behavior in the past. Part of the behavior profiles exchanged between nodes includes a description or signatures of known malicious attacks. Hence, nodes also reveal the level of awareness they may have against known exploits. During the *membership update*, each node compares ingress traffic, to its own model of input traffic as well as to the output model sent by the source of the traffic when it first requested access to the MANET. The traffic is thus tested twice to ensure it conforms to the expected profile, and that the source *did not lie* about its own behavior. Also, devices will periodically retrain their models to update them according to the new traffic exchanged. These new models should be broadcasted to all the members of the MANET and will need to be accepted by the other devices in order to renew its membership; if the model is considered to be abnormal, the device will be expelled from the MANET.

BARTER allows the MANET to self-configure, updating the membership of devices and expelling others, based on their behavior (and the changes of behavior) over time. BARTER deploys a fully self-configurable network where devices will be responsible for keeping their environment clean and safe without manual intervention. BARTER is built on top of a threshold cryptographic infrastructure that guarantees: (i) fully distributed group decision making: unless t or more devices agree on the acceptance the device will be rejected or expelled from the MANET; and (ii) secure communication: the shared certificates and shared secret keys guarantee a control over the identity of the devices and over the encryption of the communication channel.

2.1 BARTER Application and Threat scenario

MANETs are highly used in military environments [15][13][2] where there may not be any centralized or fixed infrastructure to depend on, and where communication conditions are real-time and very dynamic. In the battlefield, soldiers may be equipped with any type of handheld devices to communicate with its peers or superiors, continuously exchanging traffic and executing a diversity of applications depending on the objective of the mission. Typical behaviors in a military environment could be, for example, the exchange of maps or location coordinates, or the exchange of emails or voice over IP.

The BARTER framework addresses two important issues. First, it detects nodes that are trying to enter the MANET and whose handheld's behavior is different from the agreed *normal profile i.e.,* the behavior that the devices that are already part of the MANET agreed to accept as normal (certain type of content, certain frequency,...). In a military environment, soldiers that cannot show a good behavior in the past, will not be able to get access to the MANET, preventing them from accessing sensitive information. Second, BARTER will detect nodes that start to behave differently either because they have been compromised, or because they purposely misuse the information being exchanged in the MANET. In a military environment, if a soldier that was accepted to the MANET starts exchanging traffic with a different pattern than the allowed (the device might have been infected by a worm or stolen from the soldier), it will be caught and expelled from the MANET. If an *insider attack* happens and is not detected, there is a high possibility of information leakage. This can be avoided by having the members of the MANET screen all wireless traffic they see (and not only incoming traffic to their devices) and check that its destination is always a MANET member. If a MANET member is detected sending information to an outsider, it will be kicked out of the MANET. Any outsider wanting to receive information from a MANET member will need to gain access to the MANET first.

2.2 Behavior modeling

BARTER assumes that each device in the MANET uses an anomaly detection sensor to model its behavior. The anomaly detection sensor builds a profile of the typical behavior of the device in previous transactions. The profile can be built using the payload or other characteristics of the communication. The input model is a representation of the typical traffic a device receives from the multiple devices it communicates with. The output model is a representation of the typical traffic it sends to the other devices in its environment. Both input and output models are saved by each device as a Bloom Filter (BF) [1] (vector of 0s and 1s) in order to preserve privacy whenever the models are exchanged among devices in the BARTER framework. The way the traffic is mapped to a BF depends on the anomaly sensor used. The only requirement is that all devices use the same sensor with the same type of mapping. Models in BARTER are easily comparable, since AND operations allow us to discern how similar or different

the models (BFs) are, and OR operations allow us to merge models to represent unions of profiles. The input model should not be shared or exchanged with any other device, but saved as a secret key only available locally to the device, to avoid attackers crafting their local behaviors according to other devices' input models.

Any sensor can be used to obtain the representation of the behavior. For the purpose of this paper, we use a content-based anomaly sensor that implements an adaptation of Shanner's ideas [22], where we only consider two types of information (content): good samples (*goodS*) and bad samples (*badS*), and we only model with 3-grams. Although Shanner is more expensive than other anomaly detector sensors, we chose it because it rapidly captures the significant information of the traffic being exchanged. In MANETs, unlike in wired networks, the flux of traffic is variable, unstable, and may disappear at times. Thus, it is important to obtain precise models with the little traffic that may be available. We chose 3-grams because they are less computationally expensive than higher n-grams and because they capture well the specifics of email traffic (we provide details in Section 6). Every time a device trains its normalcy model, the content of the traffic received and sent by the device will be captured as 3-grams. We keep count of the most frequently seen 3-grams during training following Shanner's Formula (1), where the frequency W of each 3-gram i, $W(i)=F(i)\times U(i)\times A(i)$, is expressed as,

$$W(i) = log(\frac{x_i}{N_g})\times(\frac{1}{logN_g})\sum_{j=1}^{N_g}(p_{ij}(log(\frac{1}{p_{ij}})))\times(1-(\frac{1}{logL})\sum_{j}^{goodS,badS}(p_{ij}(log(\frac{1}{p_{ij}})))) \quad (1)$$

where $F(i)$ measures the frequency of occurrence of each distinct n-gram i over all the good samples N_g; $U(i)$ measures how uniformly distributed each unique n-gram i is spread among the set of good samples N_g; and $A(i)$ measures how uniformly distributed each unique n-gram i is spread across good and bad samples, L=2. We encourage the reader to check Shanner's algorithm explanation [22]. We just mention that the n-grams considered are those that are not only most frequently seen on each training unit ($F(i)$), but also across all training units ($U(i)$), and that do not appear on a set of *well known* bad n-grams ($A(i)$). The n-grams chosen are then hashed into a Bloom Filter of size 2^{13} (Section 6 shows that this size together with the 3-grams give a precise way of differentiating profiles). We assume that the devices contain an initial collection of known bad n-grams drawn from known bad models (Section 6 explains how bad models are defined). This collection will grow over time as devices in the MANET detect and agree on abnormal behaviors.

3 Bootstrap Phase

The bootstrap phase is run only once so that all initial devices agree on a definition of normal traffic based on their personal experiences. During bootstrap,

the devices broadcast their output models to all the others. Each device calculates the similarity between its own input model and the output models from the other devices, in order to compare the traffic that other devices are sending out to the traffic that itself is receiving. This similarity measure will be used in the future to either accept or reject new devices to the MANET.

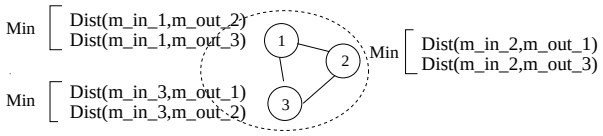


Fig. 1. Bootstrap Phase for MANET with initial members *node_1*, *node_2* and *node_3*.

As Figure 1 shows, each device *node_i* locally computes the minimum distance between its own local input model *m_in_i* and everybody else's output model *m_out_j* as *th_i=Min(Dist(m_in_i,m_out_j))* for all *j*'s in the initial MANET, where *Dist* represents the cardinality of the result vector of AND-ing the local input model Bloom Filter *m_in_i* with other device's output model BF *m_out_j*, and *Min* represents the minimum cardinality among all the cardinalities calculated of distance vectors between *m_in_i* and all other *m_out_j*. The threshold *th_i* represents the minimum requirements for a device to be accepted to the MANET by *node_i*. At the end of the bootstrap phase, each device will have a local similarity threshold *th_i* to be used for future acceptance and rejection of newcomers. Also, each device saves in a local table everybody else's output models and the similarity measure between its local input model and each output model in the table. The local threshold can be easily obtained by looking for the minimum value among the distances saved in the table. These distances can also be seen as a trust measure of the confidence that the local device has towards everybody else's models, *i.e.,* a device whose output model similarity measure to the local input model is bigger than another device's output model similarity will be more trusted by the local device, since their traffic flows are more similar. This trust value will also be used in future BARTER implementations as a way of leveraging messages coming from devices with different trust values.

The bootstrap phase is based on the assumption that the initial devices used to create the MANET represent well-behaved and well-intentioned devices, and that the devices have an accurate representation of what typical traffic in the MANET type being monitored looks like. We assume that initial behavior models for the devices at the bootstrap phase come either from training in a simulated environment, or from industry, *i.e.,* devices are sold with a built-in pre-defined model of a certain application. We also assume that the application models have a measure of the typical density (cardinality of BF) for each type of application. This *application cardinality* can be used as a countermeasure against attackers

sending out fake models containing all 1's that would trick the AND similarity calculations.

The bootstrap phase runs over a threshold cryptographic layer where a trusted dealer (TD) computes an RSA (or other type) private key to form the group secret key *SKgroup* and uses Shamir's secret sharing scheme [21] to randomly generate a polynomial f(z) over Zq of degree *t-1*, and compute each user's secret share key i as *(SKgroup,i) = f(i) mod q*. The secret key is securely transferred from the dealer to each of the users. The TD also issues a group membership certificate for each of the initial devices, which is sent securely to each of them together with the secret share, *(GMCi;SKgroup,i)*. The trusted central dealer is not used after the establishment of the ad-hoc network, leaving the system as fully distributed. From this moment on, any group of t or more devices in the MANET will be able to recover the secret using Lagrange's interpolation formula. The group distributed secret keys and certificates force the MANET to collaborate in the acceptance or rejection of devices to the MANET. A single device will not be able to generate or revoke certificates or secret keys without the help of *t-1* devices in the MANET. The merger between the cryptographic layer and BARTER implies that each of the devices in the bootstrap phase will also receive a set of individual secret and public keys, as well as an individual certificate for these keys. The individual keys and certificate will be used during communications with other devices in the MANET, to guarantee that the device is who it claims to be (using the certificate) and that the communication is encrypted, using *(SKlocal,i; PKlocal,i)*.

4 Membership Acceptance

When a new device attempts to enter the MANET, it needs to broadcast its own local output model to the MANET. The MANET members first check to see whether the device is blacklisted or not, which means that the device has already been rejected from the MANET a maximum number of times *max*; and whether the cardinality of the model is bigger than the known *application cardinality*. If it passes both checks, each device *node_i* in the MANET computes the similarity between their local input model *m_in_i* and the output model of the newcomer device *m_out_new*, as can be seen in Figure 2. Each device will then either accept or reject the newcomer depending on whether or not the similarity measure is within its local similarity threshold *th_i* calculated during bootstrap, *Dist(m_in_i,m_out_new) <= th_i* . If a minimum number of t devices agree, the device is accepted to the MANET.

If the device is accepted, all the members of the MANET send their output models to the new member. The new member saves the models together with the distances between its own input model and everybody else's output models, thus calculating its own threshold of similarity, *th_new=Min(Dist(m_in_new,m_out_j))* for all j's in the MANET, and building its own local table. The other members of the MANET save the newcomer's output model in their local tables, together with a measure of the similarity between their local input models and

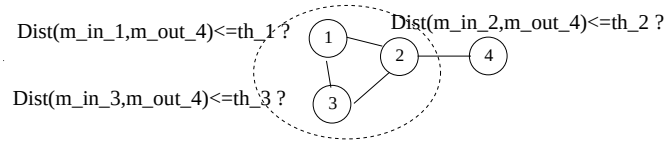


Fig. 2. Membership Acceptance of *node_4* to the MANET formed by devices *node_1*, *node_2* and *node_3*.

the newcomers' output model. The local thresholds of the existent members are unchanged since the device has been accepted based on the assumption that the local threshold *th_i* for each *node_i* is smaller than the distance between its local input model and the newcomer's output model.

If the newcomer's behavior is similar enough, each device sends its new partial shared key and its partial signature to the newcomer (following a set of rounds of information exchange [16][23]) . If the newcomer receives partial signatures and partial secret shares from at least t devices in the MANET, it will be able to compute the new GMC (Group Membership Certificate) by summing the partial signatures *modulo* q and its new share key by summing the shared keys from the other t devices. If the newcomer is not accepted by at least t devices, the devices that computed a rejection are responsible for adding the device to a grey list, which will keep track of the number of attempts by a device to enter the MANET. If a device attempts more than *max* times, it will be blacklisted. The devices that reject the newcomer, broadcast both black and grey lists to all the other members in the MANET so that all members maintain an updated common list. BARTER's decision making process is fully distributed and based on a group agreement. BARTER only allows communications to take place via a secured channel. There may be nodes that do not agree with the group's decision to reject a device and may still want to communicate with it. However, if they are running the BARTER framework, only communications via a secured channel (whose keys are given out after acceptance) are allowed, making the traffic exchange with non accepted nodes impossible unless the BARTER framework is tampered. The experiments in Section 6 are an attempt to estimate the value of t that represents a good merger point between the cryptographic layer and the BARTER layer.

5 Membership Update

The membership update represents a stage in which the MANET is already formed, and the devices that are part of it need to be continuously screened to make sure their behavior does not change drastically over time. Going back to the military environment, what happens when a device has been infected and starts sending out abnormal traffic? How can one detect that a device's profile is not normal anymore? The BARTER framework implements two different checks on the data: a model-traffic check to continuously screen the traffic received from

other devices; and a model-model check that updates the membership status of the devices based on new profiles obtained after periodic training.

5.1 Model - Traffic Security Check

Each device continuously checks the incoming traffic from any device against its local input model and against the sender's output model, which was saved when the sender was accepted. The Shanner anomaly sensor checks incoming n-grams against the local input model and the local copy of the sender's output model, and raises an alert when the number of abnormal n-grams exceeds a threshold. During communication, it may be the case that one or more devices raise an alert while receiving traffic from another device in the MANET. If this is the case, each device should send the set of n-grams that generated the alert to the devices that they trust more in the MANET, which are inferred from the trust values in the local table. Since the threshold cryptographic layer running below will need at least t devices to expel another one from the MANET, each device raising an alert will broadcast it to its t most trusted devices. If any t devices agree on the fact that it is an alert, each of them will add the source device to their local CRL (certificate revocation list) and will generate new partial secret keys and new partial certificates. These values will be broadcasted to all other devices in the MANET except for the ones in the CRL, so that the rejected device will not be able to communicate with any other device in the MANET anymore. All the devices will also update the greylist and blacklist accordingly.

As an example, Figure 3 shows a MANET with 4 devices and t with a value of 3. In the Figure, we see that *node_2*'s sensor S generates an alarm when comparing *node_4*'s traffic against its local input model *m_in_2*. This alarm is then broadcasted to the t=3 most trusted devices of *node_2*, which are *node_1* and *node_3* since *node_4* is the sender. *Node_1* and *node_3* will receive the alert n-grams from *node_2* and if their local sensors also generate an alert (as it happens), *node_4* will be greylisted or blacklisted. On the other hand, if neither *node_1* nor *node_3* raises an alert from the n-grams received, *node_4* will not be expelled from the MANET, since a minimum of t devices did not agree on the decision.

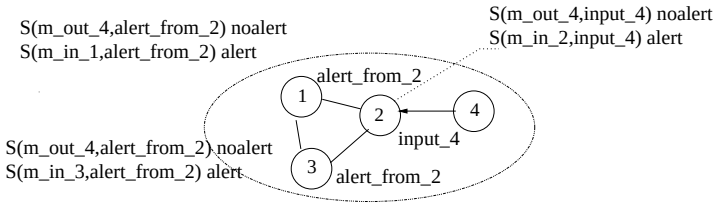


Fig. 3. Model-Traffic Security Check on traffic from *node_4* to *node_2*.

5.2 Model - Model Security Check

The anomaly sensor located within each device in the MANET, periodically retrains on the data exchanged to generate new profile models. The model retraining is scheduled to happen at different times so that no two devices generate a model simultaneously. Every time a model update happens, the new profile is broadcasted to all the MANET members. Each member will execute again the membership acceptance phase calculating the similarity distance between their local input models and the device's new output model *Dist(m_in_i,m_out_new)*. If this distance is above their normalcy threshold *th_i*, the device is rejected by *node_i*, otherwise it is be accepted. The threshold cryptographic layer under BARTER forces the MANET to generate at least t rejection responses to be able to expel the device from the MANET. Hence, if t devices in the MANET consider that the new output model does not comply with their local similarity threshold (*th_i*), the model's certificate will be revoked and added to the CRL (certificate revocation list) and a new set of shared keys and certificates will be created and shared with all the other devices in the MANET except for those in the revocation list. Each member of the MANET will also update their greylist and blacklist as well as their local tables.

6 Experiments: Description and Analysis

The main purpose of this section is to understand the performance of the merger between a threshold cryptographic layer and the BARTER layer, *i.e.,* how to add a profile exchange layer on top of a cryptographic layer to enhance access control incurring in low costs and with a good performance. With that purpose, we investigate possible values of the cryptographic threshold parameter t that was described in previous sections, and the performance of the BARTER access control in terms of devices correctly or incorrectly accepted to the MANET, or rejected from it. The cryptographic infrastructure running below the BARTER layer requires an initial number of devices to set up the distributed shared keys to secure the access control of new devices to the MANET. From a cryptographic and wireless network point of view, a lower value of t is more advantageous because it implies that in order to make any decision, any t devices in the environments of the newcomer will be enough. The bigger t is, the less probable that there will be sufficient devices to make a decision in a one-hop distance, and hence delays may occur until t decisions are gathered, maybe even forcing the newcomer to roam until it locates enough devices. On the other hand, the BARTER layer will work more accurately (with higher true rejection rate of bad profiles and lower false rejection rates of good profiles) when a high number of devices intervene in the acceptance or rejection decision of a newcomer to the MANET.

The value of t is initially defined by the cryptographic layer as the minimum number of devices that are needed to accept or reject newcomers. If during membership acceptance t never changes, the management of the MANET will be

less expensive because key regeneration and size recalculation will not be needed (key regeneration may be used but not as often), But, if t is kept fixed with its initial value, the more the MANET grows in size, the bigger the risk of a Denial-of-Service (DoS) attack is. This is due to the fact that the smaller the value of t with respect to the total size of the MANET (percentage of devices making the access control decision), the easier the MANET could be compromised and put down, *e.g.,* if t=2 for a MANET size of 100 (only 2% of the devices make the decision), the attacker would only need to compromise any two nodes in the MANET to jeopardize the ad-hoc network. Hence, we also study the performance of the system when t is updated according to the total size of the MANET during the membership acceptance phase.

This experimental section analyzes the tradeoff between the value of the cryptographic parameter t (fixed or updated) together with its related communication, regeneration and size calculation costs, and the false rejection and true rejection rates achieved by BARTER during membership acceptance for that value of t. The false rejection rate refers to the number of devices with normal behaviors that are incorrectly rejected by the BARTER layer every time a newcomer initiates the membership acceptance phase. The true rejection rate refers to the number of devices with abnormal behaviors that are correctly rejected by the BARTER layer from entering the MANET. Dynamic thresholds allow us to change the value of t according to the size of the MANET [8]. To avoid the saturation of the system due to the expensive process of modifying t, in the experiments we use an idea proposed by Narasimha *et al.* [16][23] where the value of t is recomputed only when the difference between the previous and the current processed size of the MANET is bigger than a certain *window* value.

We approximate the cost (in seconds) of the threshold cryptographic execution as follows. The cost of renewing shared keys using proactive key generation is a function of the number of devices in the neighborhood: each device regenerates a partial key and communicates it to the other devices [20]. In other words, if there are t devices in the neighborhood, the cost is the communication/spread time, otherwise the cost is increased by the time spent roaming around looking for the t devices which may not be in its neighborhood. Hence, we assume that the cost of key regeneration for a device every time a newcomer joins the MANET is proportional to the number of t devices in the group,

$$keyRegeneration(node\_i) = K \times t \tag{2}$$

where K is the factor of proportionality that also depends on the type of threshold cryptography approach used (TS-RSA, TS-DSA,...). For a fixed value of t, the total key regeneration and communication cost for the MANET can be calculated with Equation 2 at each device, but since it is done in parallel, this is also the

total MANET cost (Equation 3).

$$cost(t\_fixed) = K \times t \tag{3}$$

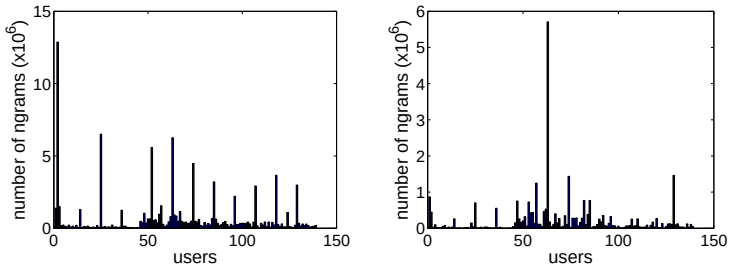
$$cost(t\_updated) = [\sum_{j=1}^{totSize-t_0/window} K \times (change + t)] + K \times t \tag{4}$$

If t changes over time, the average cost can be calculated with Equation 4 where *window* is the number of devices accepted or rejected before renewing keys, *change* is how much t is incremented or decremented, t is the number of devices that make the access control decision (t_0 is the initial value of t), and *totSize* is the final size of the MANET once all the newcomers have been accepted or rejected. The expression between square brackets represents the value of t before being incremented every time *window* newcomers have been accepted.

6.1 Testbed Description

We use as testbed a pool of 140 possible users that represent either the initial t devices in the MANET or newcomers trying to get access to the MANET. The devices are part of a real 140-node grid of devices running Debian Linux and AODV routing in the ORBIT architecture [19], a grid located in Rutgers University. The application that the devices execute in the MANET is email: each device will exchange emails with others in the MANET. We chose email because it is one of the primary applications on handhelds and a good approximation of other popular text messaging applications. To construct as realistic traffic behavior models as possible, the emails exchanged are emails from the ENRON dataset [4]. This dataset was selected because it is publicly available and experiments can be repeated by other researchers. Each device in the grid represents a user in the ENRON dataset (there are 140), running a MUA (mail user agent, mailx) and a MTA (mail transfer agent, sendmail), and exchanging emails with all or some of the other devices in the grid. Each device locally captures the input and output traffic via tcpdump on port 25 (SMTP, Simple Mail Transfer Protocol). The input and output traffic represent the emails that are received from or sent to other devices in the MANET. The devices' profiles that are going to be used in the experimental analysis are calculated using the adaptation of Shanner's algorithm [22] and the ENRON dataset. Each device gathers all of its input and output emails' content as n-grams, hashing the most frequent ones to Bloom Filters in order to build their input and output profiles.

Figure 4(a) and Figure 4(b) represent the number of input and output n-grams (3-grams) that each of the ENRON users (140) in the dataset sends and receives. As we can observe, the users are very spread in terms of number of training n-grams that are used to build each input and output behavior profile. This is a common case in MANETs where communications are unstable with devices coming in and out of the MANET, and where sensors may have very different amounts of training data. The 140 profiles that we calculate for the



(a) Input n-grams per user (b) Output n-grams per user

Fig. 4. Training Data for the Input and Output Profiles of the 140 ENRON users.

experiments, have an average cardinality (number of 1's in the Bloom Filter) of 1475 with a standard deviation of 349. Devices with more training emails will have more accurate models of what normal behavior (and normal traffic) is. For experimental purposes, the set of profiles modeled with the content of emails from the ENRON dataset (140 normal users) is referred to as *pool of normal users*. Similary, we refer to *pool of bad users*, *i.e,* profiles that should not be accepted to the MANET, as the set of profiles modeled with emails exchanging either code (Java,C) or executable files. It is important to highlight that the results presented here are limited by the fact that the profiles defined as normal (ENRON profiles) are very spread. More confined sets will definitely improve the experimental performance described.

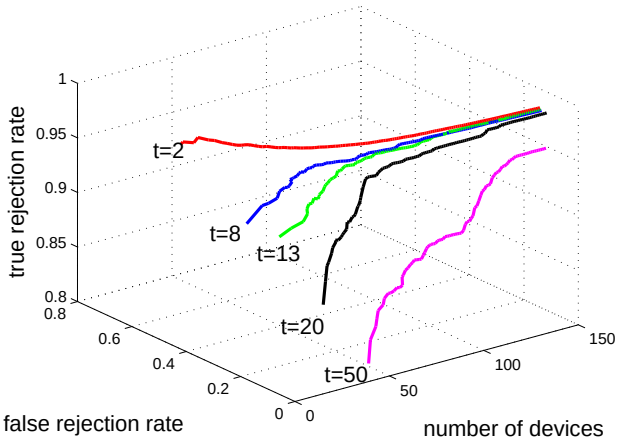
6.2 Tradeoff Analysis

We recall that t is the number of initial devices in the cryptographic system responsible for the creation of the distributed group keys, as well as the number of devices that make an acceptance or rejection decision in BARTER. We want to explore the tradeoff between BARTER's performance in terms of access control false rejection and true rejection rates, and the value of t. We analyze scenarios where t is fixed over time, and where t is updated over time. Full mesh connectivity is assumed in our experiments.

I. Value of t is fixed over time. We randomly choose sets of users' profiles from the *pool of normal users* of sizes t= 2, 8, 13, 20 and 50 respectively. These t devices first run the bootstrap phase to agree on a definition of what normal behavior is for the group, and then run the membership acceptance phase by accepting or rejecting profiles randomly chosen from the remaining profiles in the *pool of normal users* and from the *pool of bad users*. At any time, any t devices can execute the membership acceptance phase. For each possible value of t, the process of bootstrap and membership acceptance is repeated 60 times to compensate for the random selection, and averaged over all runs. As we can observe in Figure 5(a), for t=2 and t=8, 43% and 23% of the 10 first newcomers trying to enter the MANET with normal profiles are rejected (false rejection

t	FR,TR for first 10 devices	FR,TR for last 10 devices	Crypto Cost
2	0.433 0.985	0.05 0.999	$2 * K$
8	0.23 0.905	0.04 0.997	$8 * K$
13	0.103 0.950	0.035 0.998	$13 * K$
20	0.035 0.900	0.0265 0.998	$20 * K$
50	0.065 0.852	0.0286 0.964	$50 * K$

(a)



(b)

Fig. 5. (a) False Rejection (FR) and True Rejection Rate (TR) averaged over the first ten and last ten newcomers trying to enter the MANET, together with the associated cryptographic costs. At any time, any t devices make the access control decision. (b) Evolution of FR and TR for different t's every time a newcomer tries to enter the MANET.

rate), despite the fact that the costs of key generation are reasonable. For t=13 and t=20, the false rejection rates of good profiles are reduced to only 10% and 3% respectively. Obviously, this improvement in the BARTER performance comes at a cost: the key generation costs are 6 and 10 orders of magnitude bigger than for smaller values of t (calculated following Equation 3). In a bad case scenario *e.g.,* K=1, the total cost of running the membership acceptance phase for all accepted devices is 20 seconds when t=20, or approximately 0.24 seconds per device. This is to be compared with a total cost of 2 seconds when t=2, or 0.02 seconds per accepted device. Over time, as more devices are accepted, these rates improve to more acceptable rates. Third column in Figure 5(a) shows false rejection rates averaged over the last 10 newcomers which are under 5% for any value of t. We can observe that at much larger values *e.g., t=50*, there is an inflection point where the false rejection and true rejection rates are worse than

for smaller values of t. This is due to the fact that it is more difficult to initially have so many devices in agreement, which typically happens when the *pool of normal users* is very spread (like the ENRON dataset). Figure 5(b) represents the evolution of the false rejection and true rejection rates every time a newcomer tries to enter the MANET, for various values of t. As explained, larger values of t (up to the inflection point) result in better false and true rejection rates specially for the first newcomers.

Table 1 shows results for a similar experiment but with half of the initial profiles having been trained with a fixed maximum of 100 input and 100 output emails. The profiles in this dataset are more spread, in terms of number of input and output emails, than the profiles used in the initial dataset. The average cardinality for this new testbed is 1042 with a standard deviation of 616 (as opposed to 1475 and 349 in the initial dataset). There is a clear increase in the false rejection rates compared to the values seen in Figure 5(a), as well as a decrease in the true rejection rate. The larger the spread among the initial profiles *i.e.,* larger standard deviation, the worse the BARTER framework access control will perform for the same t.

t	FR,TR for first 10 devices	FR,TR for last 10 devices	Crypto Cost
2	0.654 0.885	0.064 0.952	2×K
8	0.312 0.845	0.052 0.948	8×K
13	0.191 0.910	0.041 0.958	13×K
20	0.095 0.900	0.0287 0.990	20×K
50	0.104 0.840	0.0325 0.920	50×K

Table 1. False Rejection (FR) and True Rejection Rate (TR) averaged over the first ten and last ten newcomers trying to enter the MANET, together with the associated cryptographic costs. At any time, any t devices make the access control decision. Half of the total devices have been trained with very little traffic.

II. Value of t is updated over time. If the MANET evolves into a value of t that represents either a very small or a very large portion of the total size of the MANET, the risk of a DoS attack increases. In this Section, we study the changes in BARTER's performance when t is updated over time to avoid DoS attacks. The BARTER framework has the option of updating the value of t such that

$$(change/window) \geq DecMakers \tag{5}$$

$$(t_0/window) \geq DecMakers \tag{6}$$

where *DecMakers* represents the percentage of devices relative to the total size of the MANET, that are needed to make a distributed decision for the access control to the MANET, *change* is the increment amount of t, *window* is the number of devices that has been accepted every time t is updated , and t_0 is the

t_0	FR,TR first 2*window devices	FR,TR last 2*window devices	Crypto Cost
2	0.452 0.986	0.064 0.999	1260*K
8	0.242 0.898	0.058 0.972	360*K
13	0.133 0.920	0.043 0.996	195*K
20	0.064 0.881	0.0328 0.967	200*K

(a)

t_0	FR,TR first 2*window devices	FR,TR last 2*window devices	Crypto Cost
2	0.494 0.983	0.19 0.999	2256*K
8	0.28 0.890	0.062 0.974	960*K
13	0.183 0.911	0.054 0.994	715*K
20	0.0702 0.860	0.0304 0.957	420*K

(b)

Fig. 6. (a) FR and TR averaged over the first and last 2×window devices and cryptographic cost. *DecMakers*=0.5 (50% of total devices make the access control decision). t_0= 20, 13, 8, 2 increases with *window*= 40, 26, 16, 4 and with increments of 20, 13, 8 and 2 respectively. (b) As (a) but *DecMakers*=0.95, t_0= 20, 13, 8, 2 increasing with *window*= 21, 14, 9, 3 and with increments of 20, 13, 8, and 2 respectively.

initial value of t. Equations 5 and 6 guarantee that t is incremented every time the percentage of decision makers over the total size of the MANET is smaller than *DecMakers*. $DecMakers = 0.05$ or emph$DecMakers = 0.95$ represent the extreme cases where as little as 5% or as many as 95% of the devices in the MANET are needed to make an access control decision (any value below 5% or above 95% is probably too risky for DoS attacks). We run the same experiments as in the case of a fixed t, randomly choosing t_0=2, 8, 13, and 20 initial users and implementing the access control test with the rest, but this time updating the value of t when needed. The costs of the cryptographic layer in this section are calculated using Equation 4. We explore three specific cases: (*Case 1*) we only require 5% of the devices of the MANET to make the access control decision (*DecMakers = 0.05*), (*Case 2*) we require 50% of the devices to emit the decision (*DecMakers=0.5*), and (*Case 3*) we require 95% of the devices (*DecMakers=.95*) to implement the access control. For each, the values of *window* and *change* are calculated from Equations 5 and 6 for the different values of t_0.

Case 1: For t_0= 20, 13, 8 and 2, the value of t gets updated every time *window*= 400, 260, 160 and 40 devices are accepted to the MANET. Since the ENRON dataset contains 140 users, the value of t never changes because we always have more than a 5% of devices making the access control decision. This is always the case except for t=2. In fact, the false rejection rates, true rejection rates and costs for total MANET sizes smaller than the *window* value (like the ENRON dataset), are the same as in the previous section (see Figure 5(a)). *Case 2:* In this case, the value of t is modified every time *window*= 40, 26, 16 and 4 devices are accepted for t_0= 20, 13, 8 and 2 respectively. Table 6(a) shows the false rejection and true rejection rates calculated for the first 2×*window* entries

to the MANET. We choose this size to be able to see the effects of the new access control policy once it takes place after *window* users have been accepted. As can be seen, false rejection rates, true rejection rates and costs are worse than *Case 1*. But because the decision is made by a 50% of the total devices in the MANET, it is more difficult to DoS attack the MANET. *Case 3:* This case presents the worse false rejection rates, true rejection rates and costs of all three scenarios, since 95% of the total devices are needed to make a decision (Table 6(b)). In summary, it is up to the user to choose adequate *DecMakers* and t_0 values depending on the MANET being modeled and on the level of DoS attack risk that the user is willing to accept.

Figure 7 shows the changes in BARTER's performance (evolution of the false rejection rates) as a function of *DecMakers* and t_0 for *Cases* 1 through 3. As can be seen, the larger the percentage of decision makers is: (i) the larger the false rejection rates are for any given t, since it is more difficult for the devices to agree on a decision (specially in spread datasets like ENRON); (ii) the more expensive the cryptographic layer is; and (iii) the more secure against DoS attacks the system will be because the decision making process is more distributed.

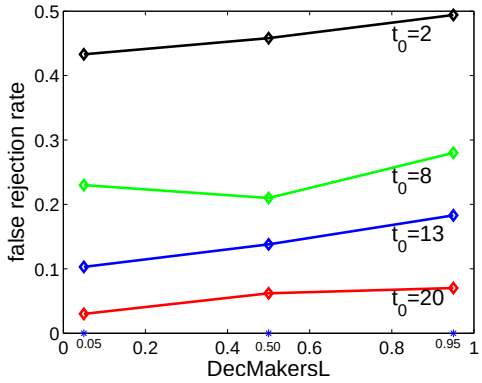


Fig. 7. BARTER's False Rejection Rate for different updated t's and *DecMakers* values. t_0 is t's initial value, incremented over time.

7 Conclusions and Future Work

We have implemented and tested BARTER, a fully distributed behavior-based access control and communication security framework for mobile ad-hoc networks. The framework allows the devices to exchange their behavior profiles in order to determine whether the behavior is similar or not to the normal behavior of the MANET and accept or reject it accordingly. BARTER's decision making is

implemented on top of a threshold cryptographic layer that guarantees a secure communication. This paper addresses the issue of how to merge appropriately both BARTER and cryptographic layers in order to achieve a good tradeoff between a) reasonable values of t to avoid expensive cryptographic computations; b) reasonable false rejection and true rejection rates during acceptance of newcomers to the MANET; and c) reasonable values of t to avoid DoS attacks.

BARTER achieves for the ENRON dataset, false rejection rates smaller than 0.03%, true rejection rates above 90% and cryptographic costs around 0.24 seconds per device accepted, for cases when approximately a 5% of the total size of the MANET is responsible for the decision making. We have also provided a discussion about reasonable t_0 and *DecMakers* values, as well as adequate measures of the spread among behavior models that will avoid DoS attacks without incrementing the cryptographic costs too much while keeping BARTER efficient. The experiments presented here can be used as a guideline to approximate the parameters t_0 and *DecMakers* for any other *pool of normal devices*.

The profiles exchanged among devices represent a model of the content shared. But content is not the unique possible representation of the behavior of a device exchanging emails. We plan to model other non-payload related parameters such as volume (number of emails) and velocity (rate at which emails are sent). Users will not only exchange models of the content they share but also histograms of usage that together with the content will give a more accurate definition of each device's profile. In addition, we will study possible ways to coordinate the membership decisions made according to payload and non-payload parameters. Thus far, we have mainly studied the membership acceptance phase, but we also plan to experiment with the membership update phase as new models are trained when more traffic is exchanged among the MANET members. We will also investigate ways to automatically learn BARTER's parameters from a pool of typical behaviors. Finally, we also intend to analyze the memory, power and bandwidth consumption of this type of framework.

References

1. Bloom B.H.: Space/Time tradeoffs in hash coding with allowable errors. *Communications of the ACM, Volume 13, Issue 7, July 1970.*
2. Cole R.: Initial Studies on Worm Propagation in MANETs for Future Army Combat Systems *Army Science Conference, September 2004.*
3. Cretu G., Parekh J., Wang K., and Stolfo S. Intrusion and anomaly detection model exchange for mobile ad-hoc networks. *In Proceedings of the IEEE Consumer Communication and Networking Conference, 2006.*
4. Enron Dataset, *www.cs.cmu.edu/ enron.*
5. Ertaul L. and Lu W. ECC Based Threshold Cryptography for Secure Data Forwarding and Secure Key Exchange in MANET(I). *In Proceedings of the International IFIP-TC6 Networking Conference, May 2005.*
6. Ferraiolo D., Cugini J. and Kuhn R. Role-Based Access Control(RBAC): Features and Motivations *In Proceedings of the 11th Annual Computer Security Application Conference, 1995.*

7. Freudenthal E., Pesin T., Port L., Keenan E. and Karamcheti V.: dRBAC: Distributed Role-based Access Control for Dynamic Coalition Environments *In Proceedings of the Twenty-second IEEE International Conference on Distributed Computing Systems (ICDCS), 2002*
8. Ghodosi H., Pieprzyk J. and Safavi-Naini R. Dynamic Threshold Cryptosystems: A New Scheme in Group Oriented Cryptography *In Proceedings of the International Conference on the Theory and Applications of Cryptology, 1996.*
9. Herzberg A., Jarecki S., Krawczyk H. and Yung M. Proactive Secret Sharing Or: How to Cope with the Perpetual Leakage *Advances in Cryptology, CRYPTO'95, 1995.*
10. IEEE Standards, http://standards.ieee.org/announcements/pr_IEEE1667_new.html
11. Kong J., Zerfos P., Luo H., Lu S. and Zhang L. Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks. *In Proceedings of the International Conference on Network Protocols, ICNP 2001.*
12. Kumar S., Raghavan V.S. and Deng J. Medium Access Control protocols for ad hoc wireless networks: A survey. *Ad Hoc Networks Journal, Volume 4(3), pp. 326-358, May 2006.*
13. Leiner B., Ruth R. and Sastry A. *Goals and Challenges of the DARPA GloMo Program, IEEE Personal Communications, December 1996.*
14. Lin A. and Brown R. The application of security policy to role-based access control and the common data security architecture *Computer Communications, Volume 23, Issue 17, November 2000.*
15. Garcia-Luna-Aceves J. *In Proceedings of the IEEE Military Communications Conference, MILCOM '97, November 1997.*
16. Narasimha M., Tsudik G. and Yi J.H On the utility of Distributed Cryptography in P2P and MANETs: the case of membership control *In Proceedings of the 11th International Conference on Network Protocols, ICNP 2003.*
17. Necula G.C and Lee, P. Safe Kernel Extensions Without Run-Time Checking. *2nd Symposium on Operating Systems Design and Implementation, OSDI'96, October 1996.*
18. Necula G.C. Proof-Carrying Code *The 24th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL97, 1997.*
19. Orbit architecture, *www.orbit-lab.org*
20. Pedersen T.P. A threshold cryptosystem without a trusted party. *In Proceedings of Eurocrypt, LNCS 547, 1991.*
21. Shamir A. How to share a secret. *Communications ACM, 22(11), 1979.*
22. Shanner. *US Patent No. 5,991,714, Nov 1999.*
23. Yi J.H and Tsudik G. Threshold Cryptography in P2P and MANETs: the Case of Access Control *Journal of Computer Networks, March 2007.*