15^{th} **ICCRTS**
"The Evolution of C2"

Selective Diffusion of Ratings in Trust Propagation for MANETs

Topics: 1. Concepts, Theory and Policy
2. Networks and Networking

Dang Quan Nguyen—Communications Research Centre
Louise Lamont—Communications Research Centre
Peter C. Mason—Defense Research & Development Canada

Point of Contact:
Dang Quan Nguyen
Communications Research Centre, Industry Canada
3701 Carling, P.O. Box 11490, Stn "H"
Ottawa, Ontario, Canada, K2H 8S2
613-949-8216
dang.nguyen@crc.gc.ca

Abstract

In mobile *ad hoc* networks (MANET), existing multicast protocols are regularly used by nodes to propagate information. The usual objective of these protocols is to efficiently deliver the information to a large number of subscribed nodes. However, in some circumstances, there may be a need to prevent some nodes from acquiring sensitive information during the broadcast. It becomes even more disruptive and challenging if we assume that nodes can collude and mutually share the information they received. We propose in this paper a key distribution and encryption scheme that allows nodes to selectively diffuse some information into the network and to prevent key sharing due to malicious nodes collusion. While the key computation and distribution are provided by a centralized server at the network configuration phase, encryption and decryption are performed by nodes in a distributed manner during the network operations. We illustrate the use of such a selective diffusion protocol with the propagation of trust ratings in a MANET. The selective diffusion of trust ratings will impact mission success by providing a mechanism to monitor the true state of the network and therefore improve users' awareness.

1 Introduction

Mobile *ad hoc* networks (MANET) consist of auto-configuring nodes that communicate using wireless equipment. Such networks are infrastructure-less, self-configured and do not require a centralized entity. These advantages make MANET suitable for critical uses such as tactical military networks, disaster recovery, etc. Messages between two out-of-range nodes are routed in a multi-hop way, through intermediate nodes by MANET's routing protocols (e.g. OLSR [1]).

One of the most studied topics in MANET is multicasting, or how to spread information into the network. Designing a MANET multicasting protocol is challenging because of the intrinsic properties of the MANETs such as frequently changing topology and nodes having limited computation capacities. Another important research topic in MANET deals with security which is a vast field that covers many subjects including authentication, data protection, etc.

In this paper, we investigate a particular problem at the intersection of MANET's multicasting and security: how to selectively broadcast a sensitive information into the network. Unlike existing multicast protocols for MANET, we focus our attention on preventing the unauthorized disclosure of sensitive broadcast information using encryption keys. Moreover, we propose an original solution to prevent malicious nodes from colluding and sharing their decryption keys. A detailed description of the problems is given in the next Section.

Although this selective broadcast protocol can be used to diffuse any sensitive information, we illustrate its use in this paper by focusing on the diffusion of trust ratings in a MANET. Because if a malicious node, supposedly intelligent, is allowed to know its own trust ratings, which are rated by its neighbors, then it may change its behaviour to appear less suspicious and avoid being detected.

The rest of this paper is organized as follows. The motivation and description of our problems of selective diffusion are presented in Section 2. We discuss the existing work related to this problem in Section 3. A light version of our encryption scheme supporting selective diffusion based on symmetric keys is presented in Section 4. We enhanced this version in Section 5 with an ability to prevent malicious nodes from colluding and sharing their decryption keys. We discuss the resilience of this encryption scheme to some common attack in Section 6. Finally, we conclude and discuss future work in Section 7.

2 Problem description

We base our work on the motivation of designing an efficient diffusion protocol of trust values in a MANET. There already exist many multicast protocols for MANET (see [2, 3, 4, 5]). They differ from our diffusion protocol in the objective of broadcasting information. While the main goal of the existing multicast protocols is to deliver the information to as many destinations as possible, our selective diffusion aims at preventing the sensitive information from being delivered to a particular node or group of nodes in the network during the information broadcast.

We justify the need of such a selective diffusion protocol by the sensitive nature of some information. For example, in a MANET enhanced with security parameters based on trust relationship, nodes continuously monitor the behaviour of their direct neighbors to calculate their trust value, a.k.a trust rating (see [6, 7, 8, 9]). Those values are then broadcasted into the network so that distant nodes can evaluate the trustworthiness of undirect neighbors. This is required to calculate any multi-hop secured route in this MANET. We will use two characters, Alice and Bob, to illustrate our problem as follows.

The problem arises when Bob's trust value, calculated by his direct neighbor Alice based on Bob's behaviour, is known to Bob when it is broadcasted into the network. If Bob is malicious then he can adjust his behaviour according to this rating to appear less threatening in the future. Later, when Bob learns that he has a good rating, he would choose to modify his behaviour, spending less efforts on maintaining the good rating without being perceived as a really bad person. This self-correcting ability can undermine any trust evaluation mechanism based on behavioural observations. We address this problem in Section 4.

This problem becomes even more challenging if we assume that malicious nodes can collude and share information between each other. From the problem above, we will design an encryption scheme such that all trust values of Bob, encrypted and broadcasted by Alice, can be decrypted by everyone except Bob. Now if Carol is another member of the network and she is also malicious then she may share her secret key with Bob so that Bob can read all his trust values just like anyone else. This problem is addressed in Section 5. Notice that our multicast protocol supporting the selective diffusion of information can be used to deliver any kind of sensitive information and not just the trust value between the members of the network.

In this paper, we assume the existence of a MANET multicast protocol capable of efficiently broadcasting information into the network, since this

is not our primary objective. We also assume the existence of a centralized server, also called *the authority*, uniquely used to authenticate nodes and distribute encryption keys to nodes at the begining of the network's operations. The encryption and decryption of information are performed by nodes in a distributed manner of the MANET. The server is not involved in these actions.

3 Related work

While musticasting is a popular research topic, the requirement of selective diffusion of broadcast information has drawn less attention from both communities of protocol designers and cryptographers.

One of the early research results on this topic can be found in [10]. According to that paper, the keys are distributed by the authority in a tamper-proof form which can not be read by the users, thus preventing their collusion and the exchange of keys. For example, the keys can be hardcoded into the hardware as well as the encryption and decryption algorithms. Furthermore, the tamper-proof module can be programmed only to output messages that satisfy a certain redundancy condition when decrypted with the correct key to avoid the attacks by invalid messages. All original messages must be provided with the redundancy condition before encryption. The drawback of this approach is that it must ensure that the keys are securely hidden in the tamper-proofed hardware.

Another solution to prevent the unauthorized disclosure of sensitive information is given in [11] where the authors address the problem of protecting a digital signature from being shown to a third party. This solution is based on a potential sanction of having an important and private information revealed along with the unauthorized disclosure of the digital signature. For example, if Alice sends a message to Bob with her digital signature and Bob reveals this signature to Carol, then it would be easy for Carol to compute Bob's private key, which is not in Bob's interest. We find that this solution appeals to the natural concept of *rational behaviour* which also applies to malicious users of the network. Therefore, we use this same concept as a justification of our protocol design in Section 5. We create a new encryption scheme in this protocol that extends this concept to the selective broadcast of information.

The selective diffusion of information can also be considered as an application of the multiple key ciphers, see [12]. In that work, the authors show that the RSA cryptosystem can be generalized to any arbitrary number of

keys. The keys are then distributed to the users in the network such that it is possible to encrypt a message intended to be decrypted by any given sub-group of users. For example, Alice can encrypt a message so that only Carol and Dave can decrypt and not Bob, or Alice can encrypt another message intended only to Dave and no one else, etc. One major advantage of this multiple-key cipher system is that we do not need 2^n different keys to address that many sub-groups of a network having n users. Only n keys are sufficient, providing they are appropriately distributed to the users and assuming that there is only one broadcast source in the network (n^2 keys are required without this assumption). However, this key scheme still does not solve the collusion problem of malicious nodes as they can mutually share keys. We present our first key scheme in Section 4, which is based only on n symmetric keys, that allows any node in the network to selectively broadcast information. Then another improvement is presented in Section 5 to prevent nodes from colluding and sharing keys.

4 Selective diffusion of ratings

We present in this section a protocol establishing symmetric keys. These keys allow nodes in the network to selectively broadcast ratings.

4.1 Keying scheme

Denote by $n_1, \ldots, n_N$ all N nodes of the network. Let k_i be a symmetric key. k_i is known to all nodes in the network except node n_i. This key is used to encrypt the rating of node n_i. Let k_B^N be another symmetric key known to all nodes $n_1, \ldots, n_N$. This last key is used to exchange information between N existing nodes so that a new incoming node (denoted as n_{N+1}) can not decrypt.

Thus, each node needs to store N symmetric keys. For example, node n_i has the following keys: $k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_N$ and k_B^N.

When node n_i wants to broadcast the ratings of its neighbour set $N(i)$, it proceeds as follows. For each neighbour $n_j \in N(i)$, the rating of n_j is encrypted with key k_j. This encrypted rating is appended into the same message along with the other encrypted ratings and the message is broadcasted by node n_i. Any node receiving this message is able to decrypt the rating of any other node except its own rating.

If a new node n_{N+1} joins the network then a new key k_{N+1} must be created and announced to N existing nodes. This announcement must not be read by node n_{N+1}. We use the pre-established broadcast key k_B^N known

to all N existing nodes to this purpose. This also leads to the necessity of a new broadcast key k_B^{N+1} to be created.

4.2 Protocol

The following steps describe the creation of these symmetric keys. We assumes that N nodes are not compromised initially. We also assume the existence of a centralized server capable of authenticating any new node that desires to join the network. This server is also responsible for establishing symmetric keys and broadcast them into the network. When a new node n_{N+1} joins the network, the following actions are taken in that order:

1. The server authenticates node n_{N+1} and ensures that it is not compromised initially. If the authentication process fails then we do not continue with the subsequence steps.

2. The server computes a new broadcast key k_B^{N+1} and send to node n_{N+1} all $N+1$ symmetric keys: $k_1, \ldots, k_N$ and k_B^{N+1}.

3. The server computes a new symmetric key k_{N+1} used to encrypt node n_{N+1}'s ratings. This key along with key k_B^{N+1} are sent to all N existing nodes. This announcement is encrypted by using key k_B^N. From this point on, key k_B^N is no longer used.

Notice that communications between the server and the new node n_{N+1} take place through a secured channel as any authentication protocol is expected to do. This can be achieved using a public-private key scheme. Thus, no stranger node can eavesdrop this conversation and obtain key k_B^{N+1}.

The next section discusses how to prevent two or more malicious colluding nodes from sharing their ratings with each other.

5 Preventing collusion

The previous keying scheme only works if nodes do not share keys with each other since the union of the key set of any two nodes suffices to decrypt all ratings in the network. In this section, we propose a mechanism to prevent nodes from sharing their symmetric keys.

Notice that if two malicious nodes n_z and n_b decide only to share the value of their ratings, everytime that value is decrypted, and not to share their symmetric keys used for the decryption then it becomes very difficult to prevent them from doing so because their exchanges can be concealed as

private user data exchanges. In fact, this attack is equivalent to the Sybil attack [13], i.e., one malicious node with multiple identities.

In our network, we assume that each node adopts a rational behaviour, even if it becomes compromised and turns malicious. By *rational behaviour*, we mean each node can keep a secret for itself and can protect this secret from being revealed to other nodes. This secret is usually the private key of the pair public-private keys. Thus, we assume that colluding malicious nodes can share most information about each other, such as ratings and symmetric keys, but they try to keep their private keys to themselves. Otherwise, if node n_z loses its private key to node n_b then node n_b will be able to impersonate n_z. Node n_z will no longer have control over the credit of its actions. Therefore, it is not rational for node n_z to reveal its private key to node n_b.

Based on this assumption of node's behavioural rationality, we construct the keys $\hat{k}_b^z$ such that if node n_z reveals key $\hat{k}_b^z$ to node n_b then node n_b can easily compute n_z's private key, denoted by x_z. Our key construction is based on ElGamal's encryption scheme [14].

5.1 Construction of undisclosable key

Let x_z, y_z be the private and public keys of a node n_z. We have:

$$y_z = \sigma^{x_z} \bmod p$$

where p is a large prime and $\sigma < p$. Both p and σ are chosen randomly.

The server then chooses a random k, such that k is relatively prime to $p-1$, and computes:

$$\begin{cases} \phi &= \sigma^k \bmod p \\ \omega &= y_b^k x_z \bmod p. \end{cases}$$

The pair (ϕ, ω), denoted by $\hat{k}_b^z$, will be the key attributed to and used by node n_z to decrypt the trust ratings of node n_b. Recall that these ratings are originated from n_b's direct neighbors.

Assume that if node n_z shows the key $\hat{k}_b^z$ to node n_b, resulting from their collusion, then node n_b can easily recover n_z's private key x_z from the following equation:

$$x_z = \omega / \phi^{x_b} \bmod p$$

since $\phi^{x_b} \equiv \sigma^{kx_b} (\bmod p)$, and $\omega/\phi^{x_b} \equiv y_b^k x_z/\phi^{x_b} \equiv \sigma^{kx_b} x_z/\sigma^{kx_b} \equiv x_z (\bmod p)$.

At this point, every node n_i in the network has a different key $\hat{k}_b^i$, $i \in \{1 \dots n\} \setminus \{b\}$ computed by the server. We will show how node n_i can use its

key $\hat{k}_b^i$ along with another key $\hat{r}_{(a,b)}^i$, also computed by the server as shown below, to decrypt the broadcasted ratings of node n_b by node n_a.

5.2 Encryption and decryption scheme

Our concern is now to develop an encryption scheme such that a node n_a can broadcast an encrypted rating of a node n_b and the other nodes are able to decrypt this rating using their key pairs $(\hat{r}_{(a,b)}^i, \hat{k}_b^i)$.

At the network configuration phase, the server randomly chooses two large numbers $M_{(a,b)}$ and $c_{(a,b)}$ such that

$$\forall i = 1, \ldots, n \text{ and } i \neq a, b : M_{(a,b)} > c_{(a,b)} \hat{k}_b^i.$$

The server then computes the following keys $\hat{r}_{(a,b)}^i$:

$$\forall i = 1, \ldots, n \text{ and } i \neq a, b : \hat{r}_{(a,b)}^i = M_{(a,b)} - c_{(a,b)} \hat{k}_b^i.$$

Each key $\hat{r}_{(a,b)}^i$ will be kept secret by node n_i like key $\hat{k}_b^i$. We will discuss the risk of disclosure of $\hat{r}_{(a,b)}^i$ later in this paper.

At the end of the network configuration phase, the server gives $M_{(a,b)}$ and $c_{(a,b)}$ to node n_a. These numbers will be kept secret by node n_a and used to encrypt ratings of node n_b during the network operations.

When node n_a wants to broadcast a rating of node n_b, it chooses a one-time number $\alpha < p$ and a random number r relatively prime to $p - 1$, with p a large prime. Node n_a then uses a symmetric encryption to encrypt the rating with key $k_b = \alpha^{M_{(a,b)}+r} \bmod p$.

Node n_a broadcasts the ciphertext along with the following information:

$$(\alpha; p; \beta; \gamma).$$

with $\beta = \alpha^{c_{(a,b)}} \bmod p$ and $\gamma = \alpha^r \bmod p$.

A node n_z, $z \neq b$, can use its key pair $(\hat{r}_{(a,b)}^z, \hat{k}_b^z)$ to recover the symmetric key k_b by doing this calculation:

$$\begin{aligned} k_b &= \left(\alpha^{\hat{r}_{(a,b)}^z} \bmod p\right) . \left(\beta^{\hat{k}_b^z} \bmod p\right) . \gamma \bmod p \\ &= \alpha^{\hat{r}_{(a,b)}^z + c_{(a,b)} \hat{k}_b^z + r} \bmod p \\ &= \alpha^{M_{(a,b)}+r} \bmod p. \end{aligned}$$

Node n_z can then decrypt the rating of n_b.

We can easily see that it is not in node n_z's best interest to disclose its secret key $\hat{r}_{(a,b)^z}$. Because when combining with node n_a's knowledges of

$M_{(a,b)}$ and $c_{(a,b)}$, the other secret key $\hat{k}_b^z$ can be computed and, along with it, n_z's private key x_z can also be revealed.

Our keying scheme relies on the difficulty of calculating discrete logarithm over a finite field: given α, p and $\alpha^x \bmod p$, the problem of computing x is believed to be intractable. Therefore, it should not be easy to compute $c_{(a,b)}$ or r from the information broadcasted by node n_a.

Notice that node n_z can compute $\alpha^{M_{(a,b)}} \bmod p$ because it is a simple product of $\alpha^{\hat{r}_{(a,b)}^z} \bmod p$ and $\beta^{\hat{k}_b^z} \bmod p$. Node n_z can then give $\alpha^{M_{(a,b)}} \bmod p$ to node n_b without the risk of revealing its secret key pair and enables n_b to compute the key k_b from subsequent broadcasts from n_a. However, since node n_a generates a new α for every broadcast, this collusion will not work.

There are some precautions node n_a must observe when choosing α. First, it must chooses α so that $M_{(a,b)}$ is not the order of α in $(\mathbb{Z}_p)^X$, i.e.: $\alpha^{M_{(a,b)}} \neq 1 \bmod p$. Otherwise, k_b is directly revealed in the broadcasted information because $k_b = \gamma$. Secondly, for the same value of prime p, the newly chosen value of α must not be a product of any number α that has already been used (and broadcasted). For example, if $\alpha_3 = \alpha_1\alpha_2$ is the chosen value of α for the current broadcast where α_1 and α_2 are the values of α that have been used in the earlier broadcasts, then one can easily deduce the value of $\alpha_3^{M_{(a,b)}} \bmod p$ from $\alpha_1^{M_{(a,b)}} \bmod p$ and $\alpha_2^{M_{(a,b)}} \bmod p$ without knowing the key pair. To prevent this from happening, node n_a can proceed with a series of decreasing values of α.

6 Mitigating Attacks

This scheme we present helps mitigate some known attacks against trust model based systems and through proper implementation can be made resilient to a number of common networking attacks. For example, many trust models have difficulty dealing with blackmail, where a malicious node threatens to ruin the reputation of another node should that node report negatively upon it. Here, the malicious node is not able to decrypt reports on its behaviour so it is unaware of what is being reported. There is no advantage to blackmail in this case. Similarly, another type of collusion in which nodes agree to report positively about one another is defused in this scheme as well, as there is no way for a node to verify that its colluding partner is carrying out its end of the bargain.

From a networking perspective, the usual types of attacks need to be considered and defended against. Since we are, in this paper, focusing on dissemination of trust values, we will consider only attacks that threaten

the trust model – impersonation, replay, and message modification attacks. Impersonation attacks can be launched only by a node that has obtained, either from the key server, another node, or by cryptographic attack, the valid keys for encrypting trust values within the network. An attack against the keys is, as mentioned in section 5.2, infeasible, and the other means of obtaining the keys depend on the security of the devices in question.

Defence against message modification and replay attacks is important, as we demonstrate by the following examples. Imagine a malicious node is behaving in a trustworthy manner and is receiving positive evaluations by its neighbours. This node could simply capture these positive ratings messages and replay them at a later time when it is concerned that it might be receiving poor ratings. Of course, replaying such messages may raise some suspicion if some of the nodes being rated in the replayed message's ratings list are no longer neighbours due to topology changes. In another, more dangerous, form of this attack, recall that the malicious node cannot decrypt and read its ratings, but it can extract and save these encrypted (and assumed positive) ratings from the ratings list it receives from and about its neighbours. At a later time when it is behaving maliciously, it can intercept ratings messages from its neighbours and swap in these saved ratings of itself in the place of the encrypted (and assumed negative) ratings. It then rebroadcasts the message with this one modification in the ratings list, leaving the ratings of all the other nodes unchanged. To prevent these attacks, standard defences can be applied around our scheme. For example, using a nonce or time stamp in the ratings messages can stop simple relay attacks, and using digital signatures based upon the public/private key pairs in our scheme can provide guarantees of ratings message integrity to prevent the interception, modification and rebroadcast attack described above.

7 Conclusion

We present in this paper an encryption scheme that allows multicast protocols to perform a selective diffusion of any sensitive information in a MANET. Our encryption scheme focuses on the requirement that the sensitive information must not be decryptable by a particular node during the broadcast. Moreover, this encryption scheme is enhanced to prevent two or more malicious nodes from colluding and sharing their decryption keys.

We illustrate the use of such a selective diffusion protocol with the propagation of trust ratings in a MANET. Because if a malicious and intelligent node is allowed to know its own trust ratings, which are rated by its neigh-

bors based on the observation of its behaviour, then this node may adapt its behaviour to appear less suspicious and avoid being detected. Therefore, our selective diffusion protocol can be used by the majority of trust evaluation methods based on behavioural observations.

Acknowledgement

This work is funded by Defence Research & Development Canada (DRDC).

References

[1] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol (olsr)," IETF RFC 3626, October 2003.

[2] J. Macker, "Simplified multicast forwarding," IEFT draft, work in progress, July 2009.

[3] Y. Lacharite, M. Wang, L. Lamont, and L. Landmark, "A simplified approach to multicast forwarding gateways in manet," in *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on*, Oct. 2007, pp. 426–430.

[4] S. Y. Cho and C. Adjih, "Optimized multicast based on multipoint relaying," in *Wireless Internet, International Conference on*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2005, pp. 42–46.

[5] S.-C. Kim and K. Shin, "A performance analysis of manet multicast routing algorithms with multiple sources," in *Software Engineering Research, Management & Applications, 2007. SERA 2007. 5th ACIS International Conference on*, Aug. 2007, pp. 73–82.

[6] L. Mui, M. Mohtashemi, and A. Halberstadt, "A computational model of trust and reputation for e-businesses," in *Hawaii International Conference on System Sciences*, vol. 7. Los Alamitos, CA, USA: IEEE Computer Society, 2002, p. 188.

[7] Y. Sun, W. Yu, Z. Han, and K. Liu, "Trust modeling and evaluation in ad hoc networks," in *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, vol. 3, Nov.-2 Dec. 2005, p. 6.

[8] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*. New York, NY, USA: ACM, 2004, pp. 1–10.

[9] D. Q. Nguyen, L. Lamont, and P. C. Mason, "On trust evaluation in mobile ad hoc networks," in *MobiSec '09: Proceedings of the 1st International ICST Conference on Security and Privacy in Mobile Information and Communication Systems*, 2009.

[10] G. J. Simmons, "How to (selectively) broadcast a secret," *Security and Privacy, IEEE Symposium on*, vol. 0, p. 108, 1985.

[11] M. Klonowski, P. Kubiak, M. Kutylowski, and A. Lauks, "How to protect a signature from being shown to a third party," in *TrustBus*, 2006, pp. 192–202.

[12] C. Boyd, "Some applications of multiple key ciphers," in *Lecture Notes in Computer Science on Advances in Cryptology-EUROCRYPT'88*. New York, NY, USA: Springer-Verlag New York, Inc., 1988, pp. 455–467.

[13] J. R. Douceur, "The sybil attack," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. London, UK: Springer-Verlag, 2002, pp. 251–260.

[14] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *Information Theory, IEEE Transactions on*, vol. 31, no. 4, pp. 469–472, Jul 1985.