12-4-2013

A Simulation-Based Study of Server Location Selection Rules in Manets Utilising Threshold Cryptography

Alastair Nisbet
AUT University of Technology, anisbet@aut.ac.nz

DOI: 10.4225/75/57b56855cd8e7
11th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia, 2nd-4th December, 2013

A SIMULATION-BASED STUDY OF SERVER LOCATION SELECTION RULES IN MANETS UTILISING THRESHOLD CRYPTOGRAPHY

Alastair Nisbet
AUT University of Technology, Digital Forensic Research Laboratories
Auckland, New Zealand
anisbet@aut.ac.nz

Abstract
Truly Ad Hoc wireless networks where a spontaneous formation of a network occurs and there is no prior knowledge of nodes to each other present significant security challenges, especially as entirely online configuration of nodes with encryption keys must be performed. Utilising threshold cryptography in this type of MANET can greatly increase the security by requiring servers to collaborate to form a single Certificate Authority (CA). In this type of CA responsibility for certificate services is shared between a threshold of servers, greatly increasing security and making attack against the CA considerably more difficult. Choosing which nodes to take on the role of a CA server can have a significant impact on the efficiency of the network, and the success of certificate requests. This research uses simulation to test different rules for choosing nodes to become servers based on their location within the network. Results show that choosing the best server location rules for particular configurations is essential in ensuring both robust security and efficient running of the network.

Keywords
MANET, network security, threshold cryptography, simulation

INTRODUCTION

A MANET is a wireless network of nodes that directly communicate with each other without messages passing through a central access point. The limited radio range of wireless communications can be overcome by nodes located between the 2 prospective communicating parties passing on the messages in 'hops' so that in theory, messages can travel unlimited distances across networks. Ad Hoc networking has significant advantages where networks are required to form quickly, often in areas where other communication infrastructure is not available. As a truly ad hoc network is entirely unplanned, generally the participating nodes will have no security configurations before network formation. Security is vital to any network to ensure that the 5 attributes that fall under the umbrella term of security can be met – confidentiality, authentication, availability, reliability and non-repudiation (Zhou and Haas 1999). Truly Ad Hoc networks present significant challenges in implementing robust security in an efficient and effective manner amongst entirely equal nodes after network formation.

All five attributes of security can be achieved to a reasonable level by a robust encryption key management system (KMS). For a MANET to utilise asymmetric encryption, at least for initial key exchange between 2 communicating parties, a key management server must be utilised to act as a certificate authority (CA). Once the server is selected, it can receive requests for encryption keys and certificates, and can verify certificates of prospective communicating parties. However, the key management server must be readily available for the creation, distribution, verification and if necessary destruction of certificates. For a very small MANET, a single server may be sufficient. However, having one node with all information about encryption keys in the network is extremely insecure. This is especially true if the MANET is a truly ad hoc network where any node may be permitted to join the network. The selection of the server at random means a dishonest node may be chosen to take on the role. If no prior behaviour of the node has been monitored, then no opportunity to notice misbehaviour exists. A dishonest server with absolute authority over the keys

for a network has authority to refuse key requests and to revoke keys by refusing to verify a certificate is valid. One method to increase availability of the CA server is to replicate the server by designating more nodes as servers who then periodically exchange their certificate information so that each server in the network contains the same information. Whilst this significantly increases availability of CA services, it correspondingly significantly reduces security as a successful compromise of any server gleans all necessary information to compromise other nodes.

The desire to provide high security, high availability of the CA and to provide redundancy should a server fail for any reason, can be implemented with the use of threshold cryptography. This can be difficult in a MANET which may be highly dynamic and have constrained devices (Alkema 2013). Threshold cryptography involves dividing the task of creating encryption keys and their corresponding certificates amongst several servers who must collaborate to perform key management tasks. Additionally, should misbehaviour of a node be reported to the CA, the same collaboration is performed to eject a node from the network. The threshold refers to the number of nodes that must collaborate. If there are *m* nodes in a network with *n* servers, then *k* servers are required for certificate services. This is written as: k<=n<m. While threshold cryptography is very effective, the efficiency of the design is not always considered. The location of the servers has shown to have a significant effect on the efficient running of the network (Yang, Zeng et al. 2007). In a MANET where public key infrastructure (PKI) is utilised, servers can be very busy issuing certificates and verifying certificates every time 2 nodes wish to communicate. For a network grows in numbers rapidly, certificate requests are frequent and should result in a successful certificate issuance to avoid repeated requests. Further, as malicious nodes can be ejected, effectively by voting to eject amongst the servers, each new conversation between nodes should be preceded by a check that the party's certificates are still valid. This means that efficiency of communications to and among the CA nodes is vital in ensuring that message passing is not hampered by continual key messages to and from servers caused by inefficiency.

EXPERIMENTAL DESIGN

The SKYE protocol was chosen as the test protocol for server location comparisons. This protocol is relatively recent and a brief description of its key management processes follows. A more thorough examination of the protocol can be found in (Nisbet and Rashid 2009).

1. 20% of the nodes in the MANET are designated as servers.
2. The threshold of server nodes is selected. This figure is tuneable for the network where security and efficiency can be traded with each other.
3. This percentage of server rules is overridden until sufficient servers are present. When a network is forming, enforcing this rule would require early nodes to wait until sufficient nodes exist to meet the threshold. For example, 20% servers with a threshold of 5 requires 25 nodes in the network before communication can begin. Therefore, this rule is overridden until there are sufficient nodes, so that in this example, the first 5 nodes will all take on server roles and another server will not be designated until there are 30 nodes.
4. A node requesting a certificate contacts the closest server. The closest is the server that is the fewest hops away.
5. The first server is responsible for contacting the other servers. Each of those servers returns their portion of the certificate to the first server.
6. The first server returns all portions of the certificate to the requester.
7. The requester generates the key pair and signature from the returned parts.
8. The first server advises all servers in the network to update their certificate list with the successful issuance of the certificate.

There are generally two approaches to this type of certificate request. The requester may request each of the *k* servers directly who then reply directly to the requester. This method is used in the MOCA protocol (Yi and Kravets 2003), However, the advantage of having a server receive a unicast

request is that the server takes responsibility for contacting the other servers. By receiving the required number of parts from the *k*-1 servers and adding its own part, the server knows that a request has been successful and can safely update the certificate list. This method is used in Zhou and Hass' unnamed protocol (Zhou and Haas 1999), and with a slight modification to the MOCA protocol in SEKM (Wu, Wu et al. 2005). If a request is sent directly from a requester to *k* servers, then essentially it is the same but with the requester taking on the role as the first contacted server.

When groups of nodes are served by a group leader alone, the mobility of the nodes can hamper efficiency. In Chauhan and Sanger's work, they found that mobility had an influence on the node that should be chosen as the group leader (Chauhan and Sanger 2012). They included the mobility of the server node, battery power and behaviour of the node when assessing the suitability to become or remain a server. Here, the mobility caused inefficiency when contact was lost meaning that the ability to exchange roles between servers and non-servers was an attribute that was vital to maintaining the efficiency within the network. If the entire network is treated as a group, then efficiency can be enhanced by looking at number of 1-hop neighbours rather than mobility alone.

One interesting approach to choosing servers is proposed by Guo, Ma, Wang and Yang. Their scheme utilises selection of nodes as servers by offering an incentive to be honest (Guo, Ma et al. 2013). Those nodes displaying the most honest behaviour are then chosen as servers. Whilst this scheme attempts to maintain efficiency of the network by ensuring honesty of servers is enforced, the selection can suffer from poorly placed servers if those are the one displaying the most honest behaviour.

The SKYE protocol is distinguished from most other MANET KMS protocols by having many tuneable parameters that can be chosen on initial formation of the network or adjusted as the network grows. The parameters manipulate settings so that efficiency and security can be adjusted to suit the application. Some applications require high security and are prepared to compromise on efficiency whilst others require high efficiency where security is less of a concern. Whilst increasing the number of servers required to form a CA increases security, the extra communication overhead as each server is added reaches a point where efficiency is severely affected with little value to security. For example, requiring 10 servers to cooperate to form a CA may give high resistance to attack. However, requiring 11 servers will make little difference but may significantly decrease efficiency. Simulations were performed to identify a threshold where performance degraded to the point that the certificate process was so inefficient as to be unworkable. This was found to be 5 servers as an acceptable maximum, however simulations were run to a value of 12 servers for the sake of thoroughness. For a network requiring relatively low security, 2 servers was the minimum required and so a range of 2 – 5 servers was found to be the effective range. Additionally, coping with node misbehaviour must be rigorously enforced as misbehaviour can occur at any level within the network (Wenjia, Joshi et al. 2010).

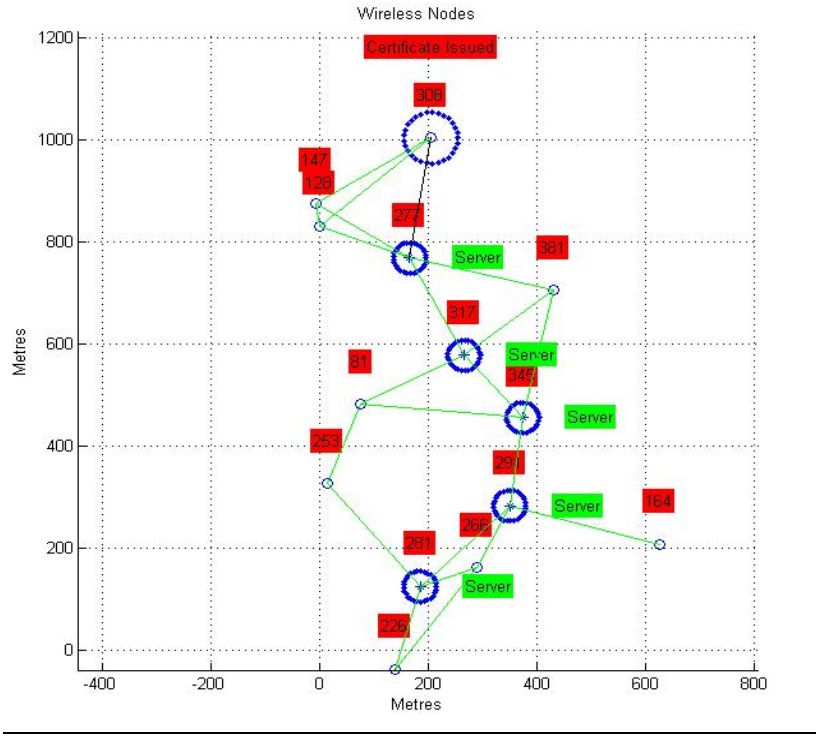

Figure 1. Certificate issuance – 5 servers required

In figure 1, the node at the top has requested a certificate from the server 1 hop away. This server has contacted 4 more servers and then returned all 5 certificate shares to the requester. The requested then assembles the shares and calculates the key pairs.

A method for evaluating security along the certificate request chain was implemented directly as that designed by the developers of MOCA. Called Composite Key Management, part of this scheme requires each node in the chain to constantly evaluate its trust in the previous node in the chain. This value begins at full trust of 1.0 and each apparent misbehavior reduces the trust by 0.1. This value can be set for the network and is tuneable for the application requirement, with a higher value adding security. A value of 0.1 would indicate considerable tolerance to perceived misbehaviour, with a value of 0.9 very strict and possibly punishing false positives to misbehaviour with complaints to the servers and eventually ejection from the network of that node. Each node takes the calculated trust value for the certificate request and uses a formula to multiply its trust value. If the server receiving the request finds the calculated trust value is below the required level, the request is ignored and a re-request must be made. Increasing security can therefore be achieved by increasing the value of *k* and increasing the trust threshold for the certificate chain.

Efficiency in the network is vital, especially as nodes join and traffic volumes grow. By carefully selecting the placement of the servers, efficiency can be significantly increased. If efficiency is not a primary concern, then this increased efficiency can be traded against adding more servers to increase security. The placement of the servers therefore has considerable benefits to the network efficiency.

Simulations were performed to compare results for different rules relating to server locations. These rules were used when the network forms and when additional servers are required to maintain the correct percentage of servers. There were 6 distinct rules implemented:

- Random: List the non-server nodes in the network and randomly choose one to become a server.
- Least: Find the non-server node with the least number of neighbours and designate that node as a server.
- Least Update: At every update period of 1 second, find the non-server node with the least number of neighbours. If it is a server, take no action. If a non-server designate that node as a server.
- Most: Find the non-server node with the most number of neighbours and designate that node as a server.
- Most Update: Find the non-server node with the most number of neighbours and designate that node as a server. If it is a server, take no action. If a non-server designate that node as a server.
- Cluster: Find a non-server node that is the neighbour of a server and designate that node as a server. In this way, servers will initially all be neighbours of each other so that a large cluster of servers forms.

The SKYE protocol was simulated using a custom built wireless network simulator written in Matlab. The results for 10 simulations were averaged to eliminate as much random variation as possible within the available timeframe, so that 66 results required 660 simulations. The hopping of messages from nodes requesting certificates to the server during inter-server communication may be affected by misbehaving nodes. Calculating the level of trust along the certificate chain and presenting the calculated trust gives confidence that a message is genuine and unaffected by misbehaving nodes. Raising the trust level required will increase security with the penalty of greater number of certificates being refused, while lowering the required level has the opposite effect. This adds a level of security within the network for certificate requests. The simulation area was set at 2.5km square to simulate a large urban area that had been rendered almost flat. This meant that wireless signals could travel equally well in all directions. Figure 2 shows the simulation area with several wireless nodes and has the minimum threshold set at 4 servers. Nodes represented as stars are servers and a single node out of range of another node has a circle drawn around it at 300 metres representing the radio range. Only nodes in networks with at least 4 servers have received certificates.

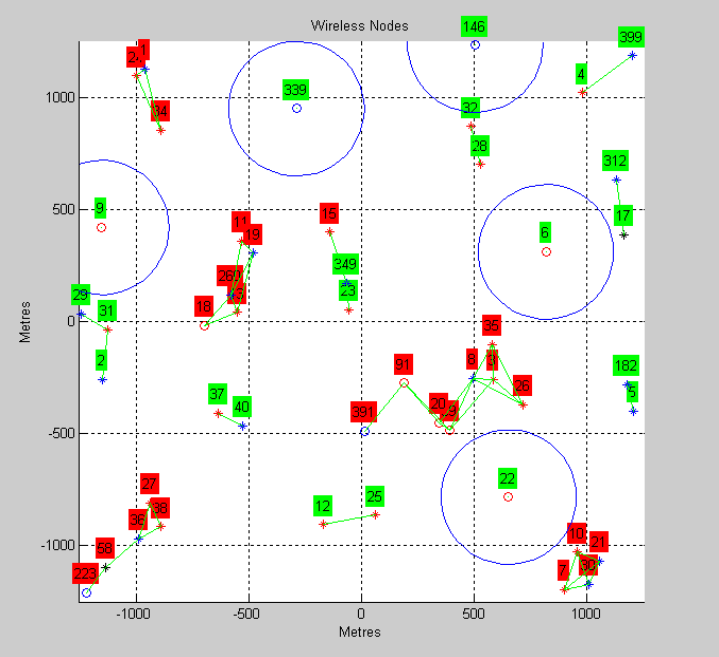

Figure 2. Wireless networks with 4 servers required

The number of likely attacks against a network will depend on the environment, members and non-members of the network and the use for the network. As a general figure, 6% of nodes were

designated as malicious nodes. These nodes regularly failed to pass on messages and had their trust by their neighbours reduced as they did so. The longer the chain that a message passes through, the higher the likelihood a malicious node would be encountered and the message or certificate request would fail to arrive. Messages were randomly sent between 2 nodes in the network giving malicious nodes frequent opportunities to misbehave. It is therefore of considerable advantage for a number of reasons for message hops from source to destination to have the shortest path possible. Any reduction in the number of hops required to reach a destination has significant benefits.

- Fewer transmissions tying up network airtime.
- Reduction in battery use of devices.
- Shorter time to accomplish the task of communication or certificate services.
- Less chance of encountering a malicious node that misbehaves with the message.
- Greater chance of successful certificate issuance.

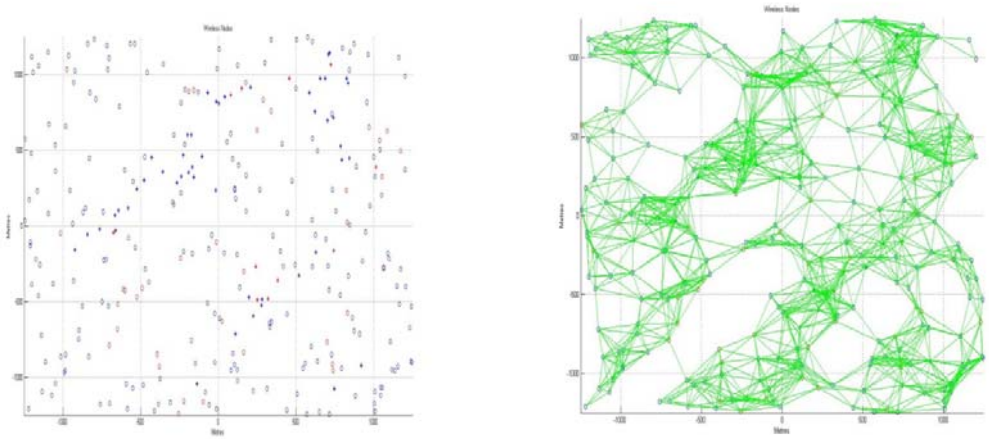

Figure 3. Wireless network with Cluster Rule with and without communication lines

Figure 3 shows 2 images of the same network which has *k* set to 5. In the first image the lines are removed to more clearly show the clustering of the servers. One large network has formed with 3 distinct clusters allowing nodes to contact a server in the cluster closest to them and all collaboration between the servers to take place within the cluster, most often with one hop neighbours. With a network protocol that requires all certificate services to be performed after network establishment, the process of requesting a certificate and successfully receiving the certificate must be as efficient as possible. Any method to increase efficiency without compromising security greatly increases the uses the protocol has, especially with a protocol such as SKYE where any node wishing to join the network is permitted to do so. The certificate issuance process utilising threshold cryptography by *k* servers cooperating, requires *k* servers to be easily contactable and that the fewest total hops for the certificate issuance can be achieved. The process adopted by SKYE utilises what the author has dubbed the Threshold Optimised MANET (TOM) to identify rules that should be in place for the server locations, dependent on the value of *k*. If *k* is set to 1, then a single server has full authority over certificate services and threshold cryptography is not employed. Therefore, the minimum value of *k* was set to 2. As the value of *k* is increased, the security of certificate issuance and revocation is increased. As a thorough comparison of the server location rules involving a wide value of *k* was desirable, experiments were conducted with *k* values ranging from 2 to 12.

RESULTS

When only 2 servers were required, the rule made little difference to the chances of success for a certificate request. In figure 4 the random plot follows approximately the centre of the 6 rules.

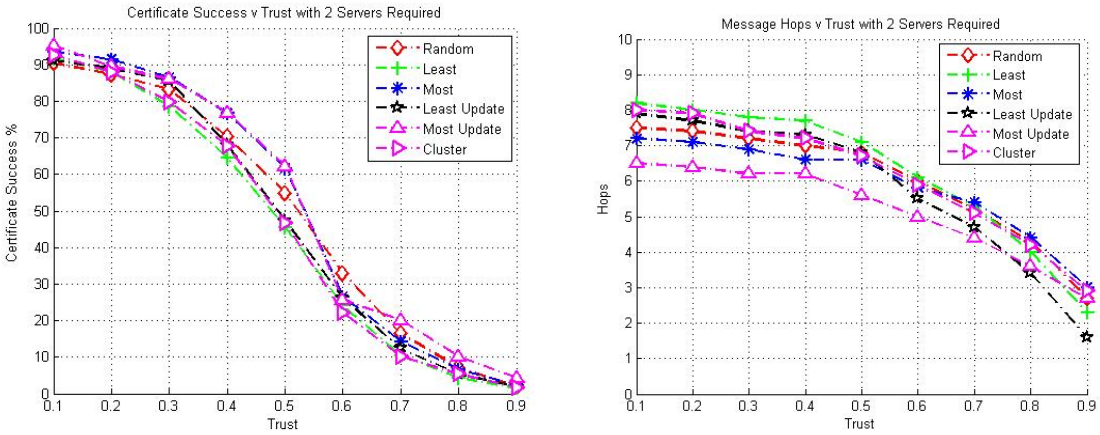

Figure 4. Success rates and hop counts with 2 servers required

As the network grows in numbers and members, spreading the nodes randomly throughout the network was both effective in terms of relatively short hops and chances of success. As the number of servers required increased, success rates drop and hop counts increase.

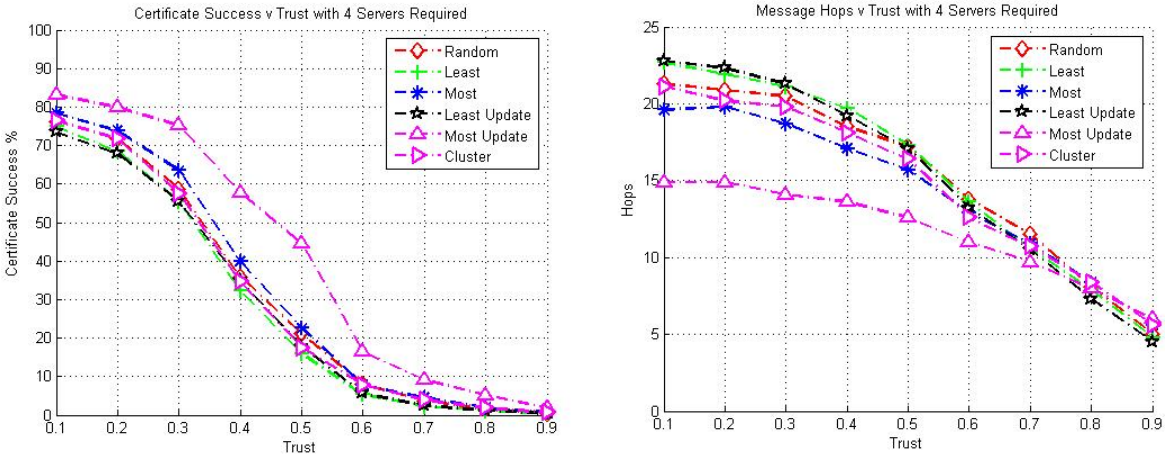

Figure 5. Success rates 4 servers required

With 4 servers required, this has changed for success rates and hops with the Most Update rule beginning to perform much more efficiently. In figure 5 the Most Update rule performs best at all trust levels but is most noticeable in the middle levels of 0.3 to 0.6. As the servers required is raised, the performance of the rules becomes much more divergent. In figure 6, results for *k* of 2 to 12 is shown and detailed results are shown in table 1.

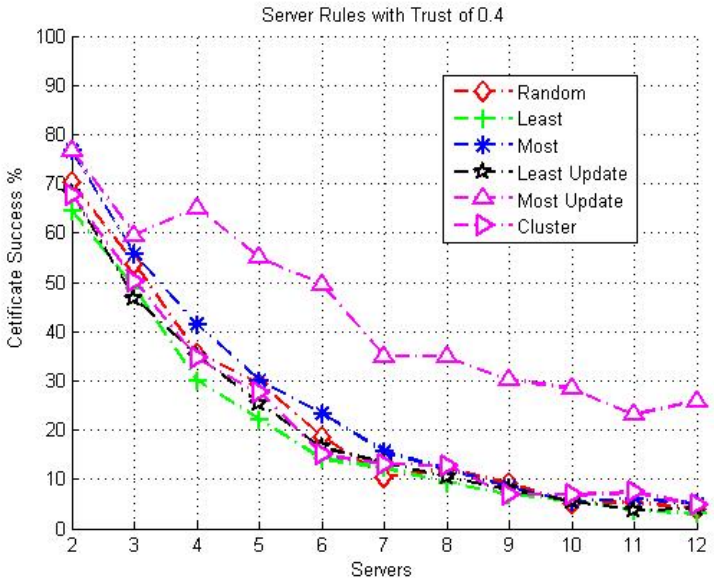

Figure 6. Success rates 2-12 servers required with 0.4 trust

With *k* of 6, the network overall is beginning to perform inefficiently requiring the trust threshold to be kept very low or the chances of a successful certificate issuance become very unlikely. Table 1 shows the certificate success rates for 2-12 server required with the trust threshold set to 0.4. Only the Most Update rule imposed above 6 servers would permit a trust threshold to be at 0.4 and then with only a 35% chance of success.

Table 1. Success Rates for Trust of 0.4

Servers	Random	Least	Least Update	Most	Most Update	Cluster
2	70	65	68	77	77	68
3	54	49	47	56	60	50
4	36	30	35	41	65	35
5	29	22	25	30	55	28
6	19	14	16	23	50	15
7	10	12	13	16	35	13
8	12	9	10	12	35	13
9	9	7	8	8	30	7
10	5	6	6	5	28	7
11	5	4	4	6	23	7
12	4	3	4	5	26	5

With a server threshold of more than 6 or a trust threshold of greater than 0.4, success rates and hop counts deteriorate to an unacceptable level. Only the Most Update rule is viable and remains viable with lower trust thresholds right up to and possibly beyond a *k* of 12. Whilst this looks encouraging, a significant drawback of updating rules is the intensive message passing required. In this case, the server location rules were checked every 1 second, and for Most Update, if a server was found to have fewer neighbors than a non-server, an exchange of roles was enforced.

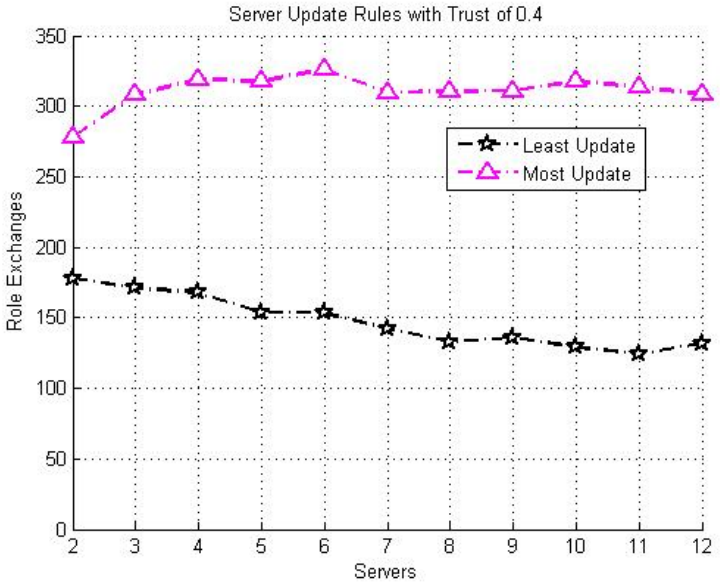

Figure 7.Server role exchanges with trust of 0.4

Figure 7 shows the number of role exchanges over the 10 minutes of simulation run time for *k* of 2 to 12. The information required to make this decision requires every node in the network to advise the servers of their number of one-hop neighbors. For a large network, this could involve hundreds of messages hopping across networks to servers every second on the chance that an exchange is necessary. Decreasing the update period would ease this slightly but at the expense of keeping the rule strictly enforced. Whilst clearly this increases efficiency, the overhead is extensive and would most likely not be warranted.

CONCLUSION

Whilst security of networks is always a consideration, security generally compromises the efficiency of the network. Generally, the higher the level of security, the more inefficient the network tends to be. This is especially true in threshold cryptography where message passing between servers is necessary for every certificate request. The results from the simulations show that efficiency can be significantly increased by choosing the correct server location rules for the network size and member numbers. The Most Update rule proves to be significantly more efficient than the others. However, the high overhead in network resources required to enforce the rule may not always be warranted. Constant swapping of roles and encryption key information between servers and non-servers reduces security and significantly increases the number of messages exchanged, meaning that unless there is an overriding reason for KMS efficiency being the priority in the network, this rule is unlikely to be commonly warranted. The next best choice is the Most rule which proves best among the other 5 rules with none of the drawbacks of the updated rules. By implementing a rule that provides high security and relatively high efficiency, the network is more useful in a greater number of applications. These results provide a guide for network administration that allows for best choices to be made with varying implementations.

REFERENCES

Chauhan, K. and A. Sanger (2012). Key Management for Group Based Mobile Ad Hoc Networks.

Ertaul, L. and N. Chavan (2005). *Security of ad hoc networks and threshold cryptography*. International Conference on Wireless Networks, Communications and Mobile Computing, 2005

Guo, Y., J. Ma, C. Wang and K. Yang (2013). "Incentive-Based Optimal Node Selection Mechanism for Threshold Key Management in MANETs with Selfish Nodes." International Journal of Distributed Sensor Networks 2013.

Nisbet, A. and M. A. Rashid (2009). A Scalable and Tunable Encryption Key Management Scheme for Mobile Ad Hoc Networks. International Conference on Wireless Networks 2009, Las Vegas, NV.

Wenjia, L., A. Joshi and T. Finin (2010). Coping with Node Misbehaviors in Ad Hoc Networks: A Multi-dimensional Trust Management Approach. Eleventh International Conference on Mobile Data Management (MDM), 2010

Wu, B., J. Wu, E. B. Fernandez and S. Magliveras (2005). Secure and Efficient Key Management in Mobile Ad Hoc Networks. Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Workshop 17 - Volume 18, IEEE Computer Society.

Yang, Y.-T., P. Zeng, y. Fang and Y.-P. Chi (2007). A Feasible Key Management Scheme in Adhoc Network. Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007.

Yi, S. and R. Kravets (2003). MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks. Annual PKI Research Workshop Program, Maryland, USA.

Zhou, L. and Z. Haas (1999). "Securing Ad Hoc Networks." IEEE Network 13(6): 24-30