www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 10 Oct 2015, Page No. 14692-14697

Detection Monitoring Of Secure Packet Transfer Over Network Traffic

Mrs.M.Shyni, Mrs.D.Ruby
PG Scholar, Department of CSE, DMI College of Engineering, Chennai.
Assistant Professor, Department of CSE, DMI College of Engineering, Chennai.

ABSTRACT

In the world of Networks, Everything on the Internet involves packets. Web page constitutes of a series of packets, and every e-mail get transfers as a series of packets. In the Proposed methodology, a monitoring system has been designed for tracing the packet transfer between the source and destination. A strategy of pattern matching has been utilized to monitor the source and destination content for its originality based on the water marking security concepts. In the proposed methodology, the monitoring system has been designed with leakage analyzer for checking the intrusion or leakage of packets between the transfers of source to destination. A security based packet tracing has been designed and the performance of the monitoring system has been visualized graphically.

Index Terms—Mobile ad hoc networks, anonymity, routing protocol, geographical routing

I INTRODUCTION

RAPID development of Mobile Ad Hoc Networks (MANETs) has stimulated numerous wireless applications that can be used in a wide number of areas such as commerce, emergency services, military, education, and entertainment. MANETs feature self-organizing and independent infrastructures, which make them an ideal choice for uses such as communication and information sharing. Because of the openness and decentralization features of MANETs, it is usually not desirable to constrain the membership of the nodes in the network. Nodes in MANETs are vulnerable to malicious entities that aim to tamper and analyze data and traffic analysis by communication eavesdropping or attacking routing protocols. Although anonymity may not be a requirement in civil oriented applications, it is critical in military applications (e.g., soldier communication). Consider a MANET deployed in a battlefield. Through traffic analysis, enemies may intercept transmitted packets, track our soldiers (i.e. Nodes), attack the commander nodes, and block the data transmission by comprising relay nodes (RN), thus putting us at a tactical disadvantage.

Anonymous routing protocols are crucial in MANETs to provide secure communications by hiding node identities and preventing traffic analysis attacks from outside observers. Anonymity in MANETs includes identity and location anonymity of data sources (i.e., senders) and destinations (i.e., recipients), as well as route anonymity. “Identity and location anonymity of sources and destinations “means it is hard if possible for other nodes to obtain the real identities and exact locations of the sources and destinations. For route anonymity, adversaries, either en route or out of the route, cannot trace a packet flow back to its source or destination, and no node has information about the real identities and locations of intermediate nodes en route. Also, in order to dissociate the relationship between source and destination (i.e., relationship un observability [1]), it is important to form an anonymous path between the two endpoints and ensure that nodes en route do not know where the endpoints are, especially in MANETs where location devices may be equipped.

Existing anonymity routing protocols in MANETs can be mainly classified into two categories: hop-by-hop encryption [2], [3], [4], [5], [6] and redundant traffic [7], [8], [9],[10], [11], [12], [13]. Most of the current approaches are limited by focusing on enforcing anonymity at a heavy cost to precious resources because public-key-based encryption and high traffic generate significantly high cost. In addition, many approaches cannot provide all of the aforementioned anonymity protections. For example, ALARM [5] cannot protect the location anonymity of source and destination, SDDR [14] cannot provide route anonymity, and ZAP [13] only focuses on destination anonymity. Many anonymity routing algorithms [3], [4], [13], [5], [6], [11], [10] are based on the geographic routing protocol (e.g., Greedy Perimeter Stateless Routing (GPSR) [15]) that greedily forwards a packet to the node closest to the destination. However, the protocol’s strict relay node selection makes it easy to reveal the source and destination and to analyze traffic.

On the other hand, limited resource is an inherent problem in MANETs, in which each node labors under an energy constraint. MANETs' complex routing and stringent channel resource constraints impose strict limits on the system capacity. Further, the recent increasing growth of Multimedia applications (e.g., video transmission) imposes higher requirement of routing efficiency. However, existing anonymous routing protocols generate a significantly high cost, which exacerbates the resource constraint problem in MANETs. In a MANET employing a high-cost anonymous routing in a battlefield, a low quality of service in voice and video data transmission due to depleted resources may lead to disastrous delay in military operations.

In order to provide high anonymity protection (for sources, destination, and route) with low cost, we propose an Anonymous Location-based and Efficient Routing protocol (ALERT). ALERT dynamically partitions a network field into zones and randomly chooses nodes in zones as intermediate relay nodes, which form a non-traceable anonymous route. Specifically, in each routing step, a data sender or forwarder partitions the network field in order to separate itself and the destination into two zones. It then randomly chooses a node in the other zone as the next relay node and uses the GPSR [15] algorithm to send the data to the relay node. In the last step, the data is broadcasted to nodes in the destination zone, providing k-anonymity to the destination. In addition, ALERT has a strategy to hide the data initiator among a number of initiators to strengthen the anonymity protection of the source. ALERT is also resilient to intersection attacks [16] and timing attacks [16].

We theoretically analyzed ALERT in terms of anonymity and efficiency. We also conducted experiments to evaluate the performance of ALERT in comparison with other anonymity and geographic routing protocols. In summary, the contribution of this work includes:

1. Anonymous routing. ALERT provides route anonymity, identity, and location anonymity of source and destination.

2. Low cost. Rather than relying on hop-by-hop encryption and redundant traffic, ALERT mainly uses randomized routing of one message copy to provide anonymity protection.

3. Resilience to intersection attacks and timing attacks. ALERT has a strategy to effectively counter intersection attacks, which have proved to be a tough open issue [16]. ALERT can also avoid timing attacks because of its no fixed routing paths for a source destination pair.

4. Extensive simulations. We conducted comprehensive experiments to evaluate ALERT's performance in comparison with other anonymous protocols.

II ALERT: AN ANONYMOUS LOCATION-BASED EFFICIENT ROUTING PROTOCOL

2.1 Networks and Attack Models and Assumptions

ALERT can be applied to different network models with various node movement patterns such as random way point model [17] and group mobility model [18]. Consider a MANET deployed in a large field where geographic routing is used for node communication in order to reduce the communication latency. The location of a message's sender may be revealed by merely exposing the transmission direction. Therefore, an anonymous communication protocol that can provide un traceability is needed to strictly ensure the anonymity of the sender when the sender communicates with the other side of the field. Moreover, a malicious observer may try to block the data packets by compromising a number of nodes, intercept the packets on a number of nodes, or even trace back to the sender by detecting the data transmission direction. Therefore, the route should also be undetectable. A malicious observer may also try to detect destination nodes through traffic analysis by launching an intersection attack. Therefore, the destination node also needs the protection of anonymity. In this work, the attackers can be battery powered nodes that passively receive network packets and detect activities in their vicinity. They can also be powerful nodes that pretend to be legitimate nodes and inject packets to the network according to the analytical results from their eavesdropped packets. The assumptions below apply to both inside and outside attackers.

1. Capabilities. By eavesdropping, the adversary nodes can analyze any routing protocol and obtain information about the communication packets in their vicinity and positions of other nodes in the network. They can also monitor data transmission on the fly when a node is communicating with other nodes and record the historical communication of nodes. They can intrude on some specific vulnerable nodes to control their behavior, e.g., with denial-of-service (DoS) attacks, which may cut the routing in existing anonymous geographic routing methods.

2. In capabilities. The attackers do not issue strong active attacks such as black hole. They can only perform intrusion to a proportion of all nodes. Their computing resources are not unlimited; thus, both symmetric and public/private key cannot be brutally decrypted within a reasonable time period. Therefore, encrypted data are secure to a certain degree when the key is not known to the attackers.

Dynamic Pseudonym and Location Service

In one interaction of node communication, a source node S sends a request to a destination node D and the destination responds with data. A transmission session is the time period that S and D interact with each other continuously until they stop. In ALERT, each node uses a dynamic pseudonym as its node identifier rather than using its real MAC address, which can be used to trace nodes' existence in the network. To avoid pseudonym collision, we use a collision resistant hash function, such as SHA-1 [19], to hash a node's.

III SYSTEM STUDY

Feasibility study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

Economic feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

System analysis

System Analysis is a combined process dissecting the system responsibilities that are based on the problem domain characteristics and user requirements.

Existing system

The conventional systems maintain high detection accuracy while coping with some of the traffic variation in the network (e.g., network delay and packet loss), however, their detection performance substantially degrades owing to the significant variation of video lengths.In this paper, we focus on overcoming this issue by proposing a novel content-leakage detection scheme that is robust to the variation of the video length. By comparing videos of different lengths, we determine a relation between the length of videos to be compared and the similarity between the compared videos. Therefore, we enhance the detection performance of the proposed scheme even in an environment subjected to variation in length of video. Through a test bed experiment, the effectiveness of our proposed scheme is evaluated in terms of variation of video length, delay variation, and packet loss.Packets loss is a major problem in the existing system.Delay in data sharing while sending packets from source to destination.

The contributions in solving these problems are:

Sender in an application, used to send the content with watermarking information. To enhance the process, we are using monitoring system to detect the Content leakage and intrusion .Once monitored the content leakage and intrusion in an application, each and every content is transferred as packet while packet transfer there is no intrusion or defect in content. It checks the originality of the content after packet transferred from a monitoring system not only the original content also with watermarked content.

Monitoring system analyzes the originality of the content and it is time consuming while transferring packets. These contents along with watermarked source send to the receiver through monitored system only.

IV PROPOSED SYSTEM

In this work, the attackers can be battery powered nodes that passively receive network packets and detect activities in their vicinity. They can also be powerful nodes that pretend to be legitimate nodes and inject packets to the network according to the analytical results from their eavesdropped packets. The assumptions below apply to both inside and outside attackers. Mobile Ad Hoc Networks (MANETs) use anonymous routing protocols that hide node identities and/or routes from outside observers in order to provide anonymity protection. However, existing anonymous routing protocols relying on either hop-by-hop encryption or redundant traffic either generate high cost or cannot provide full anonymity protection to data sources, destinations, and routes. ALERT uses the hierarchical zone partition and randomly chooses a node in the partitioned zone in each step as an intermediate relay node (i.e., data forwarder), thus dynamically generating an unpredictable routing path for a message. Exhibit consistency with our analytical results. Both prove the superior performance of ALERT in providing anonymity with low cost of overhead. Recall that anonymous routing protocols can be classified into hop-by-hop encryption and redundant traffic. Experiment results show that ALERT can offer high anonymity protection at a low cost when compared to other anonymity algorithms. It can also achieve comparable routing efficiency to the base-line GPSR algorithm. Monitoring system is there to check out the outflow in sending of packets. No delay in the packet transfer from source to destination.

System design

System Design involves identification of classes their relationship as well as their collaboration. In objector, classes are divided into entity classes and control classes. The Computer Aided Software Engineering (CASE) tools that are available commercially do not provide any assistance in this transition. CASE tools take advantage of Meta modelling that are helpful only after the construction of the class diagram. In the FUSION method some object-oriented approach likes Object Modelling Technique (OMT), Classes, and Responsibilities. Collaborators (CRC), etc, are used. Objector used the term ”agents” to represent some of the hardware and software system. In Fusion method, there is no requirement phase, where a user will supply the initial requirement document. Any software project is worked out by both the analyst and the designer. The analyst creates the user case diagram.

The designer creates the class diagram. But the designer can do this only after the analyst creates the use case diagram. Once the design is over, it is essential to decide which software is suitable for the application.

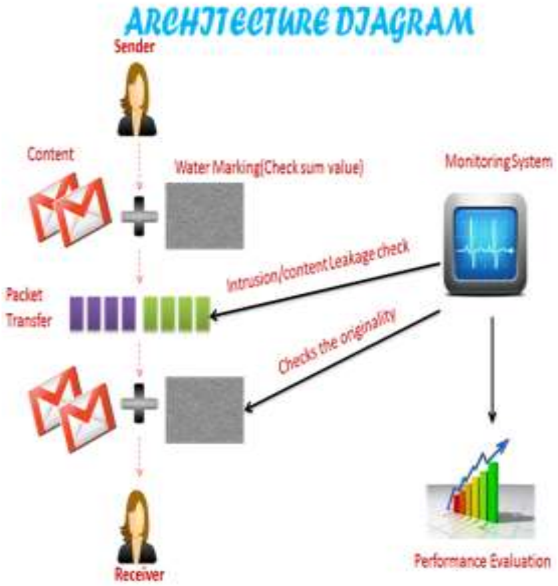

Fig: Architecture diagram

The Data Flow diagram is a graphic tool used for expressing system requirements in a graphical form. The DFD also known as the “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that to become program in system design.Thus DFD can be stated as the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail.The DFD consist of series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flows in the system. A DFD describes what that data flow in rather than how they are processed. So it does not depend on hardware, software, data structure or file organization.

V IMPLEMENTATION

Implementation module

Implementation is the stage of the project when the theoretical design is turned out into a working system.

1. User Detail Module
2. Packet Sharing Module
3. Intrusion Detection Module
4. Information Leakage Module
5. Packet Monitoring Module
6. Performance Evaluation Module

1. User Detail Module

In the user details module, the inputs from the user will be fetched and stored in the database for the standard for sending the data’s from the source to the destination.

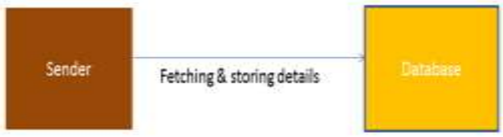

Fig : Consumer feature

2. Packet Sharing Module:

The sender will be sending the packets of data’s of information. Every data’s from the source is sent via packets to reach the destination of the receiver.

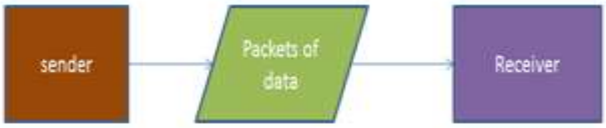

Fig: Packet distribution

3. Intrusion Detection Module:

In the Intrusion detection module, the loss of packets will be checked and evaluated based on the sent data’s of packets transfer. If the data’s are lost during packet transfer then obviously , there will be an intruder changing the content in the data packets.

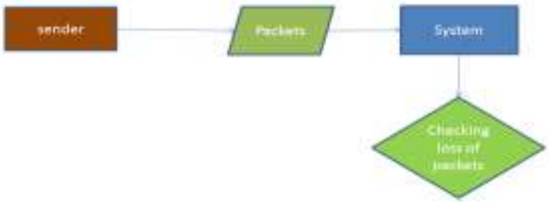

Fig: Information outpouring checks

4. Information Leakage Module

Information has been leaked or changed by the intruder or because of any other reasons will be checked in the information leakage check module .In this module, packets will be checked on the traversal of source to destination of the specified users accordingly.

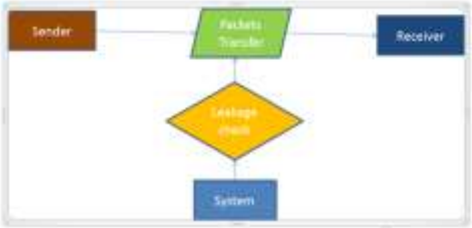

Fig: Information Leakage

5. Packet Monitoring Module

The packet Monitoring Module will be emphasized with the checking up of the data loss during the packet transfer from the sender side to the client side data exchange(Real time example of our project is “Video streaming of Data “ ,i.e. (YouTube buffering delay concepts).

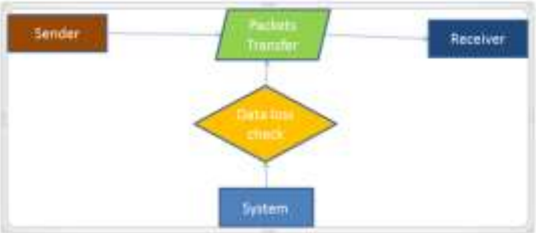

Fig: Packet Monitoring

6. Peformance evaluation module

The overall performance of our system will be checked and evaluated in the performance evaluation module based on the original packet data transfer from the user to the receiver of the traffic network scenario.

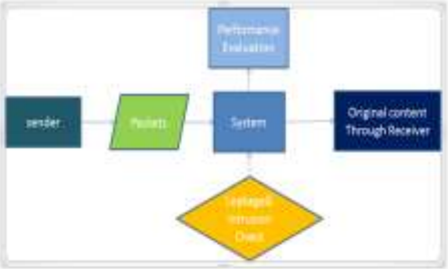

Fig: Performance Evalution

VI SYSTEM TESTING

Testing objectives

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of tests

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

System test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White box testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black box testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

All field entries must work properly. Pages must be activated from the identified link. The entry screen, messages and responses must not be delayed.

Features to be tested

Verify that the entries are of the correct format**.** No duplicate entries should be allowed**.** All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

VII CONCLUSION AND FUTURE ENHANCEMENT

ALERT further strengthens the anonymity protection of source and destination by hiding the data initiator/receiver among a number of data initiators/ receivers. It has the “notify and go” mechanism for source anonymity, and uses local broadcasting for destination anonymity. In addition, ALERT has an efficient solution to counter intersection attacks. ALERT’s ability to fight against timing attacks is also analyzed. Experiment results show that ALERT can offer high anonymity protection at a low cost when compared to other anonymity algorithms. It can also achieve comparable routing efficiency to the base-lingers algorithm. Like other anonymity routing algorithms, ALERT is not completely bulletproof to all attacks. Future work lies in reinforcing ALERT in an attempt to thwart stronger, active attackers and demonstrating comprehensive theoretical and simulation results.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their comments, which were invaluable in improving in improving this brief.

REFERENCES

[1] Y. Chu, S.G. Rao, S. Seshan, and H. Zhang, “Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture,”Proc. ACM SIGCOMM, pp. 55-67, Aug. 2001.

[2] Z. Yang, H. Ma, and J. Zhang, “A Dynamic Scalable Service Model for SIP-Based Video Conference, ”Proc. Ninth Int’l Conf. Computer Supported Cooperative Work in DE, pp. 594-599, May 2005.

[3] Y. Chu, S.G. Rao, S. Seshan, and H. Zhang, “Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture, ”Proc. ACM SIGCOMM, pp. 55-67, Aug. 2001.

[4] O. Adeyinka, “Analysis of IPSec VPNs Performance in a Multimedia Environment, ”Proc. Fourth Int’l Conf. Intelligent Environments, pp. 25-30, 2008.

[5] E.I. Lin, A.M. Eskicioglu, R.L. Lagendijk, and E.J. Delp, “Advances in Digital Video Content Protection,” Proc. IEEE,vol. 93, no. 1, pp. 171-183, Jan. 2005.

[6] S. Craver, N. Memon, B.L. Yeo, and M.M. Yeung, “Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications,”IEEE J. Selected Areas Comm.,vol. 16, no. 4, pp. 573-586, May 1998.

[7] M. Barni and F. Bartolini, “Data Hiding for Fighting Piracy,”IEEE Signal Processing Magazine,vol. 21, no. 2, pp. 28-39, Mar. 2004.

[8] K. Su, D. Kundur, and D. Hatzinakos, “Statistical Invisibility for Collusion-Resistant Digital Video Watermarking, ”IEEE Trans. Multimedia, vol. 7, no. 1, pp. 43-51, Feb. 2005.

[9] E. Diehl and T. Furon, “Watermark: Closing the Analog Hole,” Proc. IEEE Int’l Conf. Consumer Electronics,pp. 52-53, 2003.

[10] Y. Liu, Y. Guo, and C. Liang, “A Survey on Peer-to-Peer Video Streaming Systems, ” Peer-to-Peer Networking and Applications, vol. 1, no. 1, pp. 18-28, Mar. 2008.

[11] B. Zhu, Z. Wan, M.S. Kankanhalli, F. Bao, and R.H. Deng, “Anonymous Secure Routing in Mobile Ad-Hoc Networks,” Proc. IEEE 29th Ann. Int’l Conf. Local Computer Networks (LCN), 2004.

[12] A.R. Beresford and F. Stajano, “Mix Zones: User Privacy in Location-Aware Services,” Proc. IEEE Second Ann. Conf. Pervasive Computing and Comm. Workshops (PERCOMW), 2004.

[13] X. Wu, J. Liu, X. Hong, and E. Bertino, “Anonymous Geo-Forwarding in MANETs through Location Cloaking,” IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 10, pp. 1297-1309, Oct. 2008.

[14] K. El-Khatib, L. Korba, R. Song, and G. Yee, “Anonymous Secure Routing in Mobile Ad-Hoc Networks,” Proc. Int’l Conf. Parallel Processing Workshops (ICPPW), 2003.

[15] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, “Data-Centric Storage in Sensornets with GHT, a Geographic Hash Table,” Mobile Network Applications, vol. 8, no. 4, pp. 427-442, 2003.

[16] J. Raymond, “Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems,” Proc. Int’l Workshop Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability (WDIAU), pp. 10-29, 2001.

[17] T. Camp, J. Boleng, and V. Davies, “A Survey of Mobility Models for Ad Hoc Network Research,” Wireles Communications and Mobile Computing, vol. 2, pp. 483-502, 2002.

[18] X. Hong, M. Gerla, G. Pei, and C.C. Chiang, “A Group Mobility Model for Ad Hoc Wireless Networks,” Proc. Second ACM Int’l Workshop Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 1999.

[19] Debian Administration, http://www.debian-administration.org/users/dkg/weblog/48, 2012.