Secure Key Management and Verification of Mobile Ad Hoc Networks

Nazir Ahmad Zafar[1] and Ishtiaq Ahmed Choudhry[2]

[1,2]Department of Computer Science, King Faisal University, Saudi Arabia
Emails: nazafar.ichoudhry@kfu.edu.sa

Abstract: Mobile ad hoc networks (MANETs) are self-configured nodes that are connected to each other without any static infra-structure like conventional wired networks. This attractive technology offers many interesting applications in different areas such as rescue operations, and army war zones. However, MANETs are exposed to many security challenges. The conventional security solutions for wired or wireless networks are ineffective and inefficient due to cooperative nature of MANETs. A significant amount of research is found in the literature to address these security challenges. In this paper, we have reviewed many proposed security solutions. We have proposed modified security architecture to address some security issues. Then we have used formal methods to define the security architecture using Z notation. Finally we have analyzed and verified these specifications using Z/Eves toolset.

[Zafar NA and Ahmed IC. **Secure Key Management and Verification of Mobile Ad Hoc Networks** *J Am Sci* 2013;9(1):117-123]. (ISSN: 1545-1003). http://www.jofamericanscience.org. 21

Keywords: Mobile ad hoc networks, security, Formal methods, Z notation, Z/Eves

1. Introduction

The Mobile Ad Hoc Network (MANET) is an autonomous system of mobile routers and associated hosts connected by wireless links. As the routers are free to move randomly and organize themselves arbitrarily, therefore wireless topology may change rapidly and unpredictably. MANETs are typically assumed to be self-forming and self-healing [1].This is because the typical applications of such networks require nodes to form networks quickly without any human intervention. These networks have many interesting applications such as in military battlefield environment where typically many nodes need to be interconnected. Moreover, the mobility of such nodes is highly unpredictable and deployment of a fixed infra-structure may not be possible. MANETs are viewed as potential solutions providing much more support and flexibility to such an environment. Another relevant application is that of emergency response. During major emergencies and disasters such as hurricanes or large explosions, communication infrastructure in the immediate area of the disaster or emergency may be unusable, unavailable, or completely destroyed.

Despite the fact that MANET technology has dynamic infra-structure that offers many interesting applications. However, being wireless connection, MANETs are exposed to many security challenges. There are two main types of malicious attacks that could destabilize the network services. Firstly, attacks may come from malicious nodes that are not part of the network and trying to join the network without authorization. Such nodes are typically called outsiders. There are many issues that are identified due to outsider nodes and solutions are provided in [3][4][5][6][7].Secondly, there are attacks from nodes that are authorized to be part of the network and are typically called insiders. Insider nodes may launch attacks because they have been compromised by an unauthorized user through some form of remote penetration, or have been physically captured by a malicious user [8].The conventional security protocols for wired or wireless networks are incapable in securing the network because of the topology of the MANETs is changing constantly. It is necessary for each node to update routing information so as to prevent some kind of potential attacks that try to make use of vulnerabilities in the statically configured routing protocols [2].

In this paper, a review of the security architecture in place for accessing MANETs is critically analyzed. It will be focused to security attacks from the outsiders. There are mainly two types of key management techniques symmetric and asymmetric. We have modified and developed an existing key management security architecture proposed in[9][10]. The security specifications have been defined using Z notation. For this purpose, first of all, basic definitions required to define security model are provided. Then formal specification of the fundamental components is described. The components are composed to describe the mobile ad hoc network. Finally key management and security properties verification is provided. Rest of the paper is organized as follows. In section 2, literature review is presented. Modified key management algorithm is described in section 3. Formal specification is provided in section 4 following the model analysis in section 5. Finally, paper is concluded in section 6.

1. Literature Review

Cellular networks and wireless LANs (WLAN) are two very common infrastructure based networks and these networks has impelled a revolutionary change in the computing world. The concept of ubiquitous computing emerges and becomes one of the research hotspot in the computer science society [2]. In the ubiquitous computing environment, individual users wanted to use several electronic platform through which they can access all the required information whenever and wherever they may be. The demand for ubiquitous connectivity of mobile users and the consequent deployment of public networks, access control systems got much importance in the computing science [11].The SPINACH system establishes a "prison wall" which controls the flow of traffic between hosts connected to the public ports and hosts on the departmental network. Based on MAC addresses, SPINACH routers only allow the DHCP traffic, SPINACH server traffic, and authorized user traffic to go through the network. NetBar [13] separates public LANs for configuration and authentication. Traffic coming from a public port is confined with limited connectivity to the authentication and DHCP servers. Full connectivity is granted only after proper authentication. In [14], access control is implemented through the collaboration of intelligent hubs that are capable of disabling and enabling specific ports and an authentication-enhanced DHCP server, so that only authenticated clients are properly configured. Although they are effective in restricting access by unauthorized mobile users to a fixed, wired, or wireless network infrastructure, these designs do not apply in the context of MANETs where no reliable network infrastructure exists. There is no switch, router, gateway, or dedicated servers where the client traffic converges to be regulated or audited. As the network itself is composed of autonomous mobile clients, it is subject to impact due to the potential misbehaviors of any individual entity.

2. Modified Key Management Algorithm

Access-control to the network, as it is traditionally achieved by a LAN's firewall, thus becomes more difficult to deal with. The importance of key management cannot be overemphasized for mobile adhoc networks. When employing cryptographic schemes, such as encryption or digital signatures, to protect both control and data traffic, a key management service is always required. Key management is the process by which those keys are distributed to nodes on the network and how they are further updated if required, erased, and so on.

There are two main categories of cryptographic systems, namely symmetric and asymmetric. The key management process involves different techniques for these types of systems. Key management in MANETs is more difficult than in traditional networks. This is because of several factors, such as the vagaries of wireless links, lack of a central authority, constraints on resources to predetermine the neighbors of a node after deployment which is further worsened on account of the mobility of nodes in such networks.

In this paper we have focused ourselves to consider the asymmetric key management combined with the threshold cryptography (TC).The traditional approach towards developing an asymmetric key-based system is based on the use of a certification authority (CA). A public key certificate is a statement issued by some trusted party also called the certification authority, which guarantees that the public key indeed belongs to the claimed user. The trusted party (i.e. the CA) then digitally signs this statement. In order to sign certificates binding the public key of a node to the identity of the node, the CA might use a procedure that includes verifying the identity of the node and also verifying that the node has the private key corresponding to the public key. This approach is not practical in MANETs for several reasons. Firstly, a CA will be a vulnerable point in the network, especially if it is not distributed. More importantly, in order to carry out the key management operations, the CA will have to be accessible all the time. If the CA is unavailable, then the nodes in the system might be unable to update/change keys. An approach to solving this problem is to distribute the trust reposed in a single CA over a set of nodes, thereby letting the nodes share the responsibility of key management [1]. This is the approach that has emerged some asymmetric algorithms [18] [19].

In [17], the authors propose using a scheme based on the technique of threshold cryptography to distribute the private key of the certification authority. Knowledge of this key is distributed over a subset of the nodes in the network. The system, made up of the nodes in the network, is expected to have a public–private key pair. This key pair is created initially by a trusted authority before deployment of the nodes. Following that, the private key is divided into n shares using an (n, t + 1) threshold cryptography scheme. These n shares are then allocated to n arbitrarily chosen nodes by the authority that created the public–private key pair. These chosen nodes are called servers. The central authority is only needed during the bootstrapping phase. Each server also has its own key pair and stores the public keys of all the nodes in the network. In particular, each server knows the public keys of other servers. As a result, the servers can establish secure links among themselves. The service as a whole has a public–private key pair K–k. The public key K is known to all nodes while the private k is divided into shares $s_1, \ldots, s_n$, with each

server having one share. Each server also has a public–private key pair Ki–ki. Whenever a certificate has to be signed using the private key of the system, the servers are contacted. Each server generates a partial signature for the certificate using the share of the private key that the server has. The partial signature is then submitted to a combiner that computes the overall signature from the partial signatures. Note that the combiner will not be able to create the overall signature without the partial signatures.

In [10], the authors describe an approach for distributing the functions of a certification authority. The difference from [17] is that in this case any node instead of a chosen few can contain a share of the private key of the service. A new node that does not have a certificate will have to contact at least t + 1 servers. These servers can issue a certificate to such anode after establishing the identity of the node. Any t + 1 nodes can also renew a certificate. In addition, a node that does not possess a share can obtain one by contacting any group of at least t + 1 nodes that already possess the share. The bootstrapping of the system needs a trusted authority that provides the initial shares to the first t + 1 nodes.

We now propose our key management model that we have verified. For our model, we assumed that mobile ad hoc networks are sensors based in an military or natural disastrous environment. Instead of dividing private keys into many shares and each server will have one share as described by [17], we choose to have t+1 servers according to the threshold cryptography and each server have a set of private keys. The partial key generation by all servers and then combined using a combiner as presented in [17], will consume a significant amount of energy and MANETs are highly energy conscious networks. The server nodes are arbitrary and all other nodes are classified as clients. There will be only one public key that will common for all the nodes. Any new node would like to join the MANET, will provide his public-private key pair to the one of the server nodes, and server will issue a CA with his signatures to the client node. The new node can establish a communication link to any of the nodes in the network. This CA will be valid for all nodes, however, there will be a timeout period for the renewal of the CA. If any node could not get CA from one server, node will be directed to contact another server. This particular node can have (t+1)/2 number of tries to get CA. This condition will ensure that this node under consideration might be a malicious node.

3. Formal Specification

In this section, formal specification of the system and its security is described using Z notation. For this purpose, first of all, basic definitions for defining the system are provided. Then different types of nodes are described for specifying the graph under the network. Only two types of nodes, namely, client and server are considered. Server is assumed to be more powerful than client having more information and computing capability. Graph is defined based on the nodes and their relationships. The relationships among the nodes are in fact edges of the graph. Formal definition of the mobile adhoc network is provided based on the definition of the graph following the key verification mechanism. Finally, security properties are formalized based on the network. Z notation is used for the formal specification. Schema in Z is a powerful structure used for defining variables, components, encapsulation of components for defining the system, and describing the properties in terms of invariants.

4.1 Formal Specification of MANETs

In this section, formalization of the components of mobile adhoc network is given for defining the whole system. At first, formal definition of key is provided below by using the schema Key consisting of three components, namely, public, private and length. The first two variables are defined as sequence type with values as sequence of integers. The third variable is for defining maximum length of the public or private key. The schema consists of two parts in addition to name of the schema written in the first horizontal line. Definitions are given in first part and invariants are defined in second part of the schema. In the predicate part it is stated that size of public or private key is equal to length.

Key

$$public, private: \text{seq}\mathbb{N};\ length: \mathbb{N}$$

$$\#public = length \land \#private = length$$

Two type of nodes, client and server, are assumed in our model. As most of the information is same in both the nodes, hence, an abstract schema Node Info is defined to be used in the nodes. Formal definition of schema is described below. The schema consists of four components, Key, neighbors, power and status. The schema Key is defined above. The variable neighbors, type of power set of Node, represents set of neighbors of a node. The node is defined as a set type at an abstract level of specification. The variable power is used to represent energy of the battery having three values as, low, medium and high. The last one variable status shows state of the node that is active, de-active or out of range. In the predicate part of the schema, it is stated that status is de-active if power is low. Further, the status is active if the power is medium or high.

$[Node]; Status ::= ACTIVE \mid DACTIVE \mid ORANGE$

$Power ::= LOW \mid MEDIUM \mid HIGH$

NodeInfo

$Key; neighbours: \mathbb{F}\ Node$
$power: Power$
$status: Status$

$status = DACTIVE \Leftrightarrow power = LOW$
$status = ACTIVE \Leftrightarrow power = MEDIUM \lor power = \mid HIGH$

The first type of node is defined below by using the Client schema. The schema consists of four components that is node information, identifier, node type and certificate. The variable type is required because node is of type client or server. As a node requires permission before having access to the network, therefore, a variable certificate is defined to issue a certificate to the node after permission. The certificate has two values, valid or invalid certificate.

$Type ::= CLIENT \mid SERVER$

$Certificate ::= VALID \mid INVALID$

Client

$NodeInfo$
$cid: Node; type: Type; certificate: Certificate$

The second type of node is defined using the schema Server. The schema consists of five components, node information, node identifier, record about private keys, type of node and set of certificates to be issued to the nodes which need access to the network. In the predicate part, it is stated that private key must be an element of the set of certificates in the server.

Server

$NodeInfo$
$sid: Node; privates: \mathbb{F}(seq\mathbb{N})$
$type: Type$
$certificates: \mathbb{F}\ Certificate$

$private \in privates$

4.2 Formal Specification of MANETs

Formal specification of MANETs is defined after definition of model in graph theory. Then key management verification will be defined.

The security properties, authentication and authorization, will be provided in the same section. The definition of database will be described for key verification.

Now after definition of the nodes, graph relation is defined by the schema Graph. We have assumed that if there is a link between two nodes then communication is possible. The graph schema consists of three components which are clients, servers and edges. The variables, clients and servers are nodes of the graph relation whereas edges is a set of links between client to client, client to server or server to server.

Graph

$clients: \mathbb{F}\ Client$
$servers: \mathbb{F}\ Server$
$edges: \mathbb{F}(Node \times Node)$

$\forall edge: Node \times Node \mid edge \in edges$
$\bullet\ \exists n1, n2: Node$
$\mid n1 \in \{ c: Client \mid c \in clients \bullet c.cid \}$
$\lor n1 \in \{ s: Server \mid s \in servers \bullet s.sid \}$
$\quad \land n2 \in \{ c: Client \mid c \in clients \bullet c.cid \}$
$\lor n2 \in \{ s: Server \mid s \in servers \bullet s.sid \} \bullet edge = (n1, n2)$
$\forall n1, n2: Node$
$\mid n1 \in \{ c: Client \mid c \in clients \bullet c.cid \}$
$\lor n1 \in \{ s: Server \mid s \in servers \bullet s.sid \}$
$\land n2 \in \{ c: Client \mid c \in clients \bullet c.cid \}$
$\lor n2 \in \{ s: Server \mid s \in servers \bullet s.sid \}$
$\bullet\ \exists edge: Node \times Node \mid edge \in edges \bullet (n1, n2) = \mid edge$
$\{s: Server \mid s \in servers \bullet s.sid \} \cap \{ c: Client \mid c \in clients \bullet c.cid \} = \emptyset$
$\forall edge1: Node \times Node \mid edge1 \in edges$
$\bullet\ \exists edge2: Node \times Node \mid edge2 \in edges$
$\bullet\ (edge1.1, edge1.2) = (edge2.1, edge2.2)$

Invariants: (i) For any edge in the graph relation, there are two nodes making the edge, each node is either client or a server. (ii) For any two nodes there is an edge in the graph relation proving relationship between nodes and the edges. (iii) The set of identifiers of clients and servers are disjoints. (iv) As we know if a node u can communicate with another node v then v can also communicate with u. That is the graph relation is symmetric.

In MANETs, if a node is connected at one time it might be disconnected at another time. Hence, communication is only possible if the nodes are connected. A node is assumed to be active if its battery is charged and is in range in the network. Formal specification of MANET is given below based on the definition of the graph relation.

MANET

graph: Graph

$\forall client: Client \mid client \in graph . clients$
$\bullet \exists edge: Node \times Node \mid edge \in graph . edges$
$\bullet edge . 1 = client . cid \vee edge . 2 = client . cid$
$\Rightarrow client . status = ACTIVE$
$\forall client: Client \mid client \in graph . clients$
$\bullet \forall edge: Node \times Node \mid edge \in graph . edges$
$\bullet edge . 1 \neq client . cid \wedge edge . 2 \neq client . cid$
$\Rightarrow client . status = DACTIVE$
$\forall server: Server \mid server \in graph . servers$
$\bullet \exists edge: Node \times Node \mid edge \in graph . edges$
$\bullet edge . 1 = server . sid \wedge edge . 2 = server . sid$
$\Rightarrow server . status = ACTIVE$
$\forall server: Server \mid server \in graph . servers$
$\bullet \forall edge: Node \times Node \mid edge \in graph . edges$
$\bullet edge . 1 \neq server . sid \vee edge . 2 \neq server . sid$
$\Rightarrow server . status = DACTIVE$
$\forall client: Client \bullet client \notin graph . clients \Rightarrow client . status = ORANGE$
$\forall server: Server \bullet server \notin graph . servers \Rightarrow server . status = ORANGE$

Invariants: (i) For any client in the network, there is an edge and the client is an endpoint of the edge. The status of the client node is active. (ii) If a client is not endpoint of any edge then its status is disconnected. (iii) For any server, there is an edge and the server is an endpoint of the edge. The status of the server node is active. (iv) If a server is not endpoint of any edge in the graph relation then its status is deactivated. In MANTs, one node needs to be capable to identify the other node for establishment of a secure communication. For this purpose, public key is required for proving identity of a node. In our model, we have supposed that there is a certificate issuing authority which issues certificates to the eligible nodes. We know such authority might not be available to all the nodes [25] but we have taken this assumption for simplicity of the model. Encryption and decryption of public and private keys issues are not described. Formal specification of key verification is described by schema Key Verification given below following the verification properties in the predicate part.

KeyVerification

Graph

$\forall c1, c2: Client \mid c1 \in clients \wedge c2 \in clients \bullet c1 . public = c2 . public$
$\forall c1, c2: Client \mid c1 \in clients \wedge c2 \in clients$
$\bullet c1 . cid \neq c2 . cid \Rightarrow c1 . private \neq c2 . private$
$\forall s1, s2: Server \mid s1 \in servers \wedge s2 \in servers \bullet s1 . public = s2 . public$
$\forall s1, s2: Server \mid s1 \in servers \wedge s2 \in servers$
$\bullet s1 . sid \neq s2 . sid \Rightarrow s1 . private \neq s2 . private$
$\forall c: Client \mid c \in clients$
$\bullet \exists s: Server \mid s \in servers \bullet c . private \in s . privates$
$\forall c: Client \mid c \in clients$
$\bullet \exists s: Server \mid s \in servers \bullet c . certificate \in s . certificates$

Invariants: (i) The public key is same for all the client nodes. (ii) Private keys of all the client nodes must be distinct. (iii) The public key is same for all the server nodes. (iv) Private keys of server nodes must be distinct. (v) Private key of every client node is contained in some server. (vi) Certificate given to a client node must be issued by some certified authority.
The database system in MANET is assumed as a dynamic distributed, that is, each mobile node has an access to a local database system. As we have classified nodes by their capabilities. For small mobile, we used client and server is used as a powerful node with a larger share of resources. Clients have sufficient resources to cache a part of the database as well as some processing power. Formal specification of the database is provided by using the schema Database given below. Only three types of information are included which needs for defining security of the system.
The first one is registered component used to represent set of registered users. The second one is set of clients which are authenticated and defined as a function type. The last one is set of resources which are allowed to a client and is defined as a function.
Invariants: (i) Every authenticated client is a registered user. (ii) Private key of a registered user is in some server. (iii) Every client has some credentials permissible for access and use.

[*Resource*]

Database

Graph
registered: Client $\rightarrow$ seq$\mathbb{N}$
authenticated: Client $\rightarrow$ seq$\mathbb{N}$
resources: Client $\rightarrow \mathbb{F}$ *Resource*

$\forall$ *c: Client; key:* seq$\mathbb{N}$ | (*c, key*) $\in$ *authenticated*
• (*c, key*) $\in$ *registered*
$\forall$ *c1: Client; key:* seq$\mathbb{N}$ | (*c1, key*) $\in$ *registered*
• $\exists$ *c2: Client* | *c2* $\in$ *clients*
• *c1 = c2* $\wedge$ ($\exists$ *s:Server* | *s* $\in$ *servers* • *key* $\in$ *s . privates*)
$\forall$ *c: Client* | *c* $\in$ dom *registered* • *c* $\in$ dom *resources*

Authentication is a process which needs to be completed to show identities before communication between two nodes. Authentication is required because if an unauthorized node comes in and uses the available resources within the network it will cause a serious problem, particularly, when the node is malicious user. Therefore, it is needed a mechanism for preventing such unknown users to become a part of the network. The authentication property is described by using the schema authentication given below. The schema has three input variables in addition to the database. The first one is client which needs an access to communicate with another node. The last two variables are public and private keys. The definitions of variables are given in first part and security property is defined in second part of the schema.

Authentication

Database
client?: Client
pubkey?, prikey?: Key

---

client? $\in$ dom *registered*
$\exists$ *s: Server* | *s* $\in$ *servers*
• $\exists$ *key1:* seq$\mathbb{N}$ | *key1* $\in$ *s . privates* • *prikey? . private* = *key1*
$\exists$ *s: Server* | *s* $\in$ *servers*
• $\exists$ *key2:* seq$\mathbb{N}$ | *key2 = s . public* • *pubkey? . public = key2*
$\exists$ *s: Server* | *s* $\in$ *servers*
• $\exists$ *certi: Certificate* | *certi* $\in$ *s . certificates*
• *client? . certificate = certi* $\wedge$ *client? . certificate = VALID*
authenticated = authenticated $\cup$ {(*client?* $\mapsto$ *prikey? . private*)}

A strong access control procedure is needed to prevent identity theft of node in mobile networks. The authorization is required to increase trust level of the user after authentication. The formal specification of the authorization property is described below by using the schema Authorization. The schema consists of database, client which needs an access to resources and requested resources. In the predicate part, it is stated that client must be authenticated. If the requested resources are allowed then resources of the client must be updated by the requested resources.

Authorization

Database
client?: Client
resource?: $\mathbb{F}$ *Resource*

---

client? $\in$ dom *authenticated*
resources = resources $\cup$ {(*client?* $\mapsto$ *resource?*)}

4. Model Analysis

There does not exist any computer tool which may assure about complete correctness of a formal model. That means even the specification is written well, it may cause potential errors.

An art of writing formal specification does not provide guarantee about correctness of the model. If the specification is analyzed with computer tools, it improves confidence by identifying errors if exists in the model.

Z/Eves is one of the powerful tools used in this research for analyzing the formal specification of authentication, authorization and key verification of mobile ad hoc networks. All schemas of the model are analyzed and checked to be correct. Summary of the results is presented in Table 1 given below. Name of the schema is given in first column of the table. The symbol "Y" the table indicates that all schemas are proved correct automatically. Domain checking, reduction and proof by reduction are represented in columns 3, 4 & 5 respectively. The symbol "NA" in 4th column is used that reduction was not required in any schema on the predicates and, hence, the formal specification is proved to be written well and meaningful.

Table 1. RESULTS OF MODEL ANALYSIS

Schema Name	Syntax Type Check	Domain Check	Reduction	Proof
Key	Y	Y	NA	Y
NodeInfo	Y	Y	NA	Y
Client	Y	Y	NA	Y
Server	Y	Y	NA	Y
Graph	Y	Y	NA	Y
MANET	Y	Y	NA	Y
Key Verification	Y	Y	NA	Y
Database	Y	Y	NA	Y
Authentication	Y	Y	NA	Y
Authorization	Y	Y	NA	Y

5. Conclusion and Future Work

Mobile ad hoc network is a complex, security critical and complex natured system. It is challenging task to address security issues providing good quality of service because of the nodes mobility. In this paper, formal specification of the mobile ad hoc network, security properties and key management is described using Z notation.

Basic definitions of the system were provided before formalizing security and key management issues. Different types of nodes, client and server, increased complexity of the graph under the mobile ad hoc network. In the graph three types of edges were introduced because of the communication links between client to client, client to server and server to server. Formal specification of the mobile adhoc network is provided based on the definition of the graph. The information required for key management and verification is presented by a schema in terms of a database. The Z notation is used for the formal specification and model analysis is provided by Z/Eves toolset. We observed that schema in Z was an effective and powerful structure for defining and encapsulation of the components of the system. The predicate part of schema was used to describe the properties in terms of invariants introducing a good modeling approach at an abstract level of specification. Z is used because it has a rigorous computer tool support addressing complexity of the system. The Z/Eves facilitated us increasing confidence for consistent description of the system.

This work is part of our ongoing project on addressing and verifying securities issues of the MANETs. The secure modified algorithm will appear after completion of the project.

REFERENCES

1. Farooq Anjum, and Petros Mouchtaris, Security for Wireless Ad Hoc Networks, John Wiley & Sons, Inc., Publication, 2007.
2. W. Li and A. Joshi, Security issues in mobile ad hoc networks—a survey, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County.
3. Y. Zhang and W. Lee, Intrusion Detection in Wireless Ad-hoc Networks, in Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000), pages 275–283, Boston, Massachusetts, August 2000.
4. Jim Parker, Anand Patwardhan, and Anupam Joshi, Detecting Wireless Misbehavior through Cross-layer Analysis, in Proceedings of the IEEE Consumer Communications and Networking Conference Special Sessions (CCNC'2006), Las Vegas, Nevada, 2006.
5. Amitabh Mishra and Ketan M. Nadkarni, Security in Wireless Ad Hoc Networks. In: The Handbook of Ad Hoc Wireless Networks, CRC Press LLC, 2003.
6. Yongguang Zhang and Wenke Lee, Security in Mobile Ad-Hoc Networks. In: Ad Hoc Networks Technologies and Protocols, Springer, 2005.
7. Panagiotis Papadimitraos and Zygmunt J. Hass, Securing Mobile Ad Hoc Networks. In: The Handbook of Ad Hoc Wireless Networks, CRC Press LLC, 2003.
8. N. Borisov, I. Goldberg, and D. Wagner, Intercepting mobile communications: the insecurity of 802.11, in Proc. ACM MOBICOM, 2001, pp. 180–189.
9. H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks, IEEE/ACM Transactions on Networking, December 2004, pp. 1049–1063.
10. J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, Providing Robust and Ubiquitous Security Support for Mobile Ad Hoc Networks, in Proceedings of Ninth International Conference on Network Protocols (ICNP), November 2001.
11. M. Weiser, The Computer for the Twenty-First Century, Scientific American, September 1991.
12. G. Appenzeller, M. Roussopoulos, and M. Baker, User-friendly access control for public network ports, in Proc. IEEE INFOCOM, 1999, pp.699–707.
13. E. A. Napjus. NetBar—Carnegie Mellon's Solution to Authenticated Access for Mobile Machines. [online]http://www.net.cmu.edu/docs/arch/netbar.html
14. D. L. Wasley. (1996) Authenticating Aperiodic Connections to the Campus Network. [Online] http://www .ucop.edu/irc/wp/wp_Reports/wpr005/wpr005_Wasley.html
15. W. Aiello, S. M. Bellovin, M. Blaze, R. Canettia, J. Ioannidis, A. D. Keromytis, and O. Reingold, Efficient, DoS-Resistant, Secure Key Exchange for Internet Protocols, in Proc. ACM Computer and Communications Security Conference, Washington, DC, USA,2000, pp. 48–58.
16. L. Zhou and Z. Haas, Securing Ad Hoc Networks, IEEE Network, 13(6), 24–30 (1999).
17. H. Deng and D. P. Agrawal, TIDS: threshold and identity based security scheme for wireless ad hoc networks, Ad Hoc Networks, 2(3), 291–307 (2004).
18. A. Khalili, J. Katz, and W. A. Arbaugh, Toward Secure Key Distribution in Truly Ad-hoc Networks, in Proceedings of IEEEWorkshop on Security and Assurance inAd-Hoc Networks, 2003.
19. N.Arora and R.K.Shyamasundar. UGSP:Secure Key Establishment Protocol for Ad-hoc Networks. eds, Distributed Computing and Internet Technology: Lecture Notes in Computer Science, Volume 3347 of Springer, Berlin, 2005, pp. 391–399.
20. I. A. Choudhry, N. A. Zafar, and M. Zahrani, " Validating Statics of Long Term Evolution Mobile Communications System", Proceedings of the 2012 International Conference on Modeling, Simulation and Visualization Methods, pp. 201-206, Las Vegas, 2012.
21. Nazir A. Zafar, Umbreen Tajammul and Nabeel Sabir, "Formal Specification of Call Setup Procedure of UMTS Communications System", Proceedings of International Conference on Communication Software and Networks (ICCSN09), China, 2009.
22. Nazir A. Zafar and Umbreen Tajammul, "Handover Based Formal Dynamic Model of UMTS Communication System", Proceedings of 4th International Computer Engineering Conference, pp. 1-6, Egypt, 2008.
23. Saima Tariq and Nazir A. Zafar, "Modeling of Wireless Mobile Communication System Using Z Notation", Proceedings of the 1st International Conference on Computer, Control and Communication (IC4), Pakistan, 2007.

12/25/2012