San Jose State University
SJSU ScholarWorks

Master's Projects | Master's Theses and Graduate Research

1-1-2009

Versatile Extensible Security System for Mobile Ad Hoc Networks

Jung Chang
San Jose State University

Follow this and additional works at: http://scholarworks.sjsu.edu/etd_projects

Recommended Citation
Chang, Jung, "Versatile Extensible Security System for Mobile Ad Hoc Networks" (2009). *Master's Projects.* Paper 86.

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Versatile Extensible Security System for Mobile Ad Hoc Networks

---

Student: Jung Ting Chang
Advisor: Melody Moh, Professor
Department of Computer Science
San Jose State University

May 2009

Approval Form

Jung T. Chang has passed the defense for the project CS 298 – Versatile & Extensible Security system

_____ _____

(Moh, Melody - Professor) Date

_____ _____

(Moh, Teng - Professor) Date

_____ _____

(Stamp, Mark - Professor) Date

Acknowledgment

I want to thank my parents who nurtured me and stood by my decision throughout my college years. I also want to deliver my sincere thanks to my advisor, Dr. Melody Moh, the diligent and patient professor who supported and encouraged me throughout this hard time. Finally, I like to thank my untiring professors – Dr. Matt Bishop, Dr. Chen-nee Chuah, Dr. Teng Moh, and Dr. Mark Stamp (in alphabetical order by last name), whose great lectures and teachings inspired my works.

Versatile & Extensible Security System for Mobile Ad hoc Networks

By: Jung T. Chang

Abstract

Mobile Ad hoc Network (MANET) is becoming more and more popular in scientific, government, and general applications, but security system for MANET is still at infant stage. Currently, there are not many security systems that provide extensive security coverage for MANET. Moreover, most of these security systems assume nodes have infinite computation power and energy; an assumption that is not true for many mobiles. Versatile and Extensible System (VESS) is a powerful and versatile general-purpose security suite that comprises of modified versions of existing encryption and authentication schemes. VESS uses a simple and network-efficient but still reliable authentication scheme. The security suite offers four levels of security adjustments base on different encryption strength. Each level is designed to suit different network needs (performance and/or security), and the security suite allows individual end-to-end pair-wise security level adjustments; a big advantage for highly heterogeneous network. This versatility and adjustability let each pair of talking nodes in the network can choose a security level that prioritize either performance or security, or nodes can also choose a level that carefully balance between security strength and network performance. Finally, the security suite, with its existing authentication and encryption systems, is a framework that allows easy future extension and modification.

(Word Counts: 206)

List of Acronyms

- AAA: Authentication, Authorization, and Accounting
- AES: Advance Encryption Standard algorithm
- AK: Authorization Key
- AODV: Ad-hoc On-demand Distance Vector
- AP: Access Point
- CA: Centralized Authority / Central Authority
- CBR: Constant Bit Rate traffic
- CREP: Certificate Reply
- CREQ: Certificate Request
- DES: Data Encryption Standard algorithm
- DoS: Denial-of-Service
- EAP: Extensible Authentication Protocol
- EAP-AKA: Extensible Authentication Protocol with Authentication and Key Agreement
- EAP-SIM: Extensible Authentication Protocol with Subscriber Identity Module
- EAP-TLS: Extensible Authentication Protocol with Transport Layer Security
- GTK: Group Transient Key
- ISP: Internet Service Provider
- Kbps: Kilo bits per second
- MANET: Mobile Ad-hoc Networks
- Mbps: Mega bits per second
- MK: Master Key
- PKD: Proactive Key Distribution
- PKD-A4WH: Proactive Key Distribution with Anticipated 4 Way Handshake
- PKD-IAPP: Proactive Key Distribution with Inter-Access Point Protocol
- PMK: Pair-wise Master Key
- PTK: Pair-wise Transient Key
- RSA: Ron **R**ivest, Adi **S**hamir, and Leonard **A**dlement encryption algorithm
- SAODV: Secure AODV

- SKEXP: Symmetric Key Expiration notice
- SKREP: Symmetric Key Reply
- SKREQ: Symmetric Key Request
- TEK: Transport Encryption Key
- UDP: User Datagram Protocol
- VESS: Versatile and Extensible Security System
- WSN: Wireless Sensor Network

Table of Contents

I. Introduction 9

II. Background and Related Studies 10

2.1 Distributed Public Cryptography 10

2.2 Security in Wireless Network 12

2.2.1 Fast and Secure Handover for 802.11 12

2.2.2 Fast and Secure Handover in WiMAX 15

2.2.3 Secure Ad-hoc On-demand Distance Vector 17

III. Versatile and Extensible Security System 17

3.1 Authentication Scheme 18

3.2 Encryption Scheme 22

3.2.1 Open Mode 25

3.2.2 Lightweight Mode 26

3.2.3 Strong Mode 27

3.2.4 User Mode 28

3.2.5 Data-integrity mode 29

3.3 Security Analysis 30

3.3.1 Authentication 30

3.3.2 Open Mode 31

3.3.3 Lightweight Mode 31

3.3.4 Strong Mode 31

3.3.5 User Mode 32

3.3.6 Data-integrity Mode 32

IV. Performance Evaluation 33

4.1 Throughput 34

4.1.2 Throughput – Authentication Effect 36

4.1.3 Throughput – Encryption Effect 37

4.2 Delay 39

4.2.1 Delay – Authentication Effect 40

4.2.2 Delay – Encryption Effect 41

4.3 Data-integrity Mode Performance 43

V. Conclusion 44

5.1 Future work and possible extension 44

5.2 Concluding Remarks 44

VI. Reference 45

Appendix A: Network Simulators 49

I. Introduction

Mobile Ad-hoc Networks (MANET) is one of the most popular types of next generation wireless network. MANET allows rapid deployment of network at relatively low costs, and it is highly scalable compare to conventional infrastructure network. With these advantages, it is easy to see why MANET is gaining popularity in business, government, academic, and scientific communities. However, MANET is not without its share of disadvantages, and, at the moment, immature security system is one of them.

Many current MANET security systems only protects one of the three vital security components (confidentiality, integrity, or availability), and they do not take into account of the frequent re-connection nature that happens in node mobility. Moreover, most of them often assume mobiles have powerful processor and network has huge bandwidth available; an assumption that is often false in Wireless Sensor Network (WSN) and some low-cost MANETs. Some of the MANET security schemes are also so specialized on a single network protocol such that they are difficult, if not impossible, to be modified to work with other network protocols. Long story short, there is not yet, to the best of my knowledge, a single extensible security system for MANET that takes into the account of node's mobility and resource-limitation and is general enough to work with most MANET protocols.

Versatile and Extensible Security System (VESS) is exactly designed to meet these challenges. VESS protects both confidentiality and integrity aspects of the MANET. The digital-certificate-based authentication system, used by VESS, utilizes the popular idea of pre-authentication. The idea of pre-authentication, as used by many proposed authentication systems for 802.11 and 802.16 networks, moves bulk of re-authentication process to initial network-entry authentication. With pre-authentication, re-authentication, which happens during re-connection, is faster and easier since most of the required process are already done during node's initial authentication into the network.

VESS offers four levels of encryption system and each level place heavier emphasis on either security, network performance, or a balance of both. Individual pair of network nodes can choose the level of encryption that is best suited for their needs. In a heterogeneous network, VESS offers powerful nodes access to the best encryptions there is while allows the more

resource-limited nodes to prioritize performance over security. Authentication and encryption systems combine into a base framework that allows future developers to extend VESS with more functionalities and modifications. For example, a module to protect the availability aspect of security can take advantage of the digital certificate systems in VESS. With digital certificate system already in place, the new module can better defend against deadly Denial-of-Service (DoS) attacks [11] (more of this is discussed in section 5).

In this project, VESS is presented as a design that works on the popular Ad-hoc On-demand Distance Vector (AODV) protocol, and uses simulation to demonstrate the worst and best case performance impact from security overheads. The rest of this report is organized as following: section 2 includes the discussion on some of the existing idea that was modified and incorporated into VESS. Section 3 presents VESS designs in detail. Section 4 shows the analysis on the simulation results, and, finally, section 5 concludes the report with some remarks and possible future works.

II. Background and Related Studies

There are two aspects of security covered by VESS – secrecy and authenticity of information. The first sub-section presents an important study to address information secrecy – distributed public cryptography.

2.1 Distributed Public Cryptography

Since the invention of Caesar's Shift-Key Cipher over 2000 years ago, cryptography has come a long way in making information obscure to unauthorized viewer. Modern symmetric encryption and public-key encryption are relatively secure [19], but these encryptions, by themselves, are not useful in MANET environment. Symmetric encryptions require all nodes in MANET to obtain the encryption key before entering the network. This type encryption is secure if encryption keys are refreshed often [19], but it is not scalable and entire network encryption breaks down when a node is compromised. Public-key encryption is scalable and highly resistant against compromised nodes [19]. However, this type of encryption, alone, requires a centralized authority to securely sign and distributes public keys, and such authority cannot exist inside MANET without introducing the risk of single-point-failure.

Distributed public-key cryptography is a type of public-key cryptography specifically designed to work in distributed environment like MANET [10, 17]. The idea behinds distributed public key cryptography is that the duty of centralized authority is distributed among all the nodes in the MANET. Every node in MANET acts like a mini-centralized authority that helps distribute and verify the public keys of other MANET nodes. A popular implementation for this type of encryption is the "On-demand Distributed Public-Key Management"

On-demand Distributed Public-Key Management is a reactive public key cryptography that works independently of routing protocol [10]. There are two versions of this cryptography – static key binding and dynamic key binding. In static key binding, every node in the network has obtained some certificates (public-private key pair with identity information attached) before entering the network. These certificates are signed by a Centralized Authority (CA) *outside* the network , and this outside central authority has a public-key well known to all the nodes in the network. When a node enters the network, the signature of its certificate is verified by its neighbors using the well-known public key of the CA. If the verification fails or if the identity in the certificate belongs to a known compromised node, the neighbors refuse connection, and the node with that certificate is kept outside the network.

When two nodes wish to communicate with each other securely, these two nodes exchange public keys with each other using a method that successive verify and sign packet in each network hop. This method, called certificate chaining, requires the key exchange packets be signed by each hop and the signature verified by the next hop. This successive chain of authenticity vouching and verification allows the original public keys be safely transported to the destination. Details about certificate chaining is discussed in VESS designs in section 3.

The second version of distributed public-key cryptography is dynamic key binding. Dynamic key binding allows nodes in the network to generate its' own identity and key pairs without a third-party authority outside the network [10]. This distributed public-key cryptography is quite difficult and resource-consuming, and it is currently still being studied.

2.2 Security in Wireless Network

Authentication protocols are vital components in maintaining the integrity of any network. Although distributed public cryptography allows node to verify with each other, distributed public cryptography alone offers very bare bone authentication. Therefore, it is important to study authentication and other security protocols employed by other mobile or ad hoc networks such as 802.11 and WiMax. Since MANET places heavy emphasis on node's mobility, an authentication that is both SECURE and FAST is necessary. This section presents some studies on fast and secure authentication in 802.11 and WiMAX

2.2.1 Fast and Secure Handover for 802.11

Although handover no direct relationship with integrity, authentication is an important part of handover process. In MANET, nodes frequently move around the network and handover to other nodes. Thus, it is important to study how existing security proposals for other wireless network, such as 802.11, can achieves fast and secure handover. In 802.11 wireless network, the network can be made very secure by deploying 802.11i security protocol. 802.11i is based on Extensible Authentication Protocol of Transport-Layer Security (EAP-TLS), which is based on 802.1X. For detail information on EAP-TLS and 802.1X, please refer to [8, 9]. This section will only go over what is necessary to understand fast and secure handover for 802.11.

EAP-TLS is one of the most secure authentication network protocol, and there is currently no known practical method to break this authentication protocol (there are some theoretical methods, but they are largely infeasible in reality) [9]. Figure 1 shows EAP-TLS handshakes, detailed information is included below:

- mobile and network authentication server shares a secret symmetric key called "Master Key", and mobile and network authentication server each has their own "public key" embedded in digital certificate
- Authentication Server and mobile first exchange initiate and identify each other using EAPOL-Start, EAPOL-Req(Id), and EAPOL-Resp(Id).
- After identification, mobile and server exchange certificates and a few nonces, and agree on the supported symmetric cipher and supported symmetric cipher key length. In figure 1, this is all done using the EAP-TLS() message.

- Using the "Master Key" and one or more nonce(s) obtained in previous message exchange, mobile and authentication server independently derives a symmetric key called "Pair-wise Master Key" (PMK). This key is retained between mobile and authentication server ONLY, and the key length is decided by the authentication server during the exchange of "Master Key"
- Encrypted under PMK, mobile send an empty EAP-TLS message to server. In figure 1, this is the EAP-TLS(EAP-TLS:Empty) message. Server then send the "Access Authorized" message to mobile, encrypted using PMK. This is the EAP-SUCCESS message in figure 1.
- In figure 1, the follow message exchanges are called "EAPOL-Key Message()", but more detail are the following…
    - Using PMK and a few nonce exchanges encrypted under the new PMK, authentication server and mobile generates a new symmetric key called "Pair-wise Transient Key" (PTK).
    - Authentication server sends PTK to Access Point (AP), and, from there on, AP communicates with mobile using PTK.
    - AP and mobile exchanges a few more nonce. Using these nonce and PTK, AP and mobile can generates a new symmetric key called "Group Transient Key" (GTK).
    - All the data transmission is encrypted using GTK, and mobile and AP periodically compute and agree on new GTK to keep the freshness of the encryption.

In the original 802.11i, every time mobile handover to another AP, all these computational-intensive and time-consuming steps have to be re-done. In this sense, 802.11i sacrifices the seamlessness and speed of handover to achieve better security.

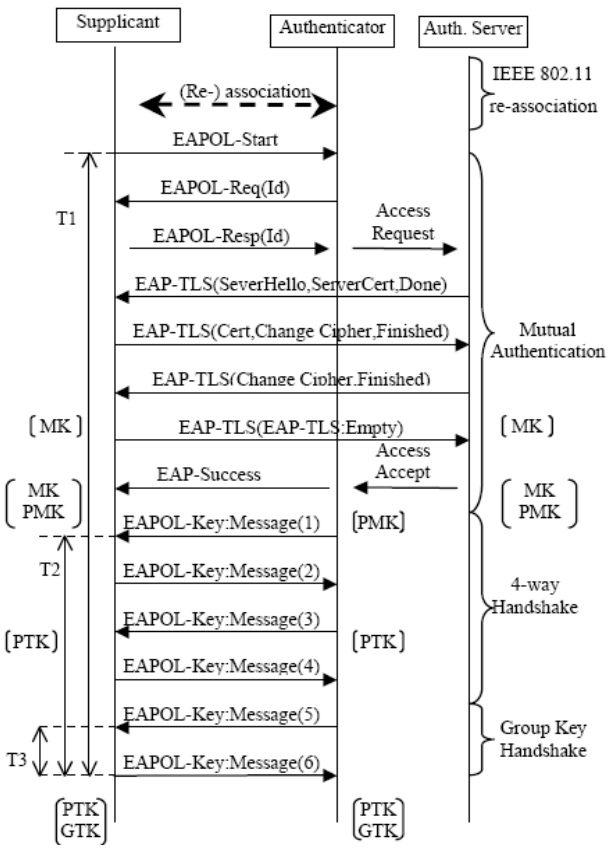

Figure 1: 802.11i authentication process. Source: [9]

This is fine for 802.11-based network where handover does not occur very frequent. However, speed and seamlessness is a big issue in MANET where mobiles frequently leaves/joins the network. MANET needs an authentication protocol of equivalent security to 802.11i but does not sacrifice the speed of authentication. Protocols proposed for fast and secure 802.11 handover can help achieve this.

Existing fast and secure handover proposes 3 methods in manipulating 802.11i such that the re-authentication during handover is fast and seamless [9]. First and the easiest method is the popular Proactive Key Distribution (PKD). Using PKD, in the initial authentication when mobile first enters the network, mobile establishes all the necessary keys with authentication server and the AP.

The authentication server gives the mobile and AP a list of neighboring APs. Using that list and with current AP's help, mobile establishes PTKs with neighboring APs (a PTK for each AP).

This stage is called "pre-authentication" stage, a stage where some keying information is established beforehand so that, during handover, less keys need to be established [8, 9]. With PKD, mobile only needs to establish GTK with the neighboring AP that the mobile is moving to; since PMK and PTK are all established and stored in neighboring AP before handover. There are less message exchanges with only GTK needed to be established, and, hence, faster handover.

The second and the slightly faster method is "PKD with Inter-Access Point Protocol" (PKD-IAPP). In PKD-IAPP, mobile, in initial authentication when first enter the network, establishes PTK with neighboring APs as well. During handover, the GTK with current AP is transferred to the neighboring AP that the mobile is moving to, and mobile can immediately resume transmission using the old GTK. After handover is completed and data transmission has started with the neighboring AP, neighboring AP becomes current AP for the mobile. Mobile and this new current AP can then negotiate a new GTK for security, all while data transmission is in process. This method ensures handover is completely seamless and brings no interruption to data transmission.

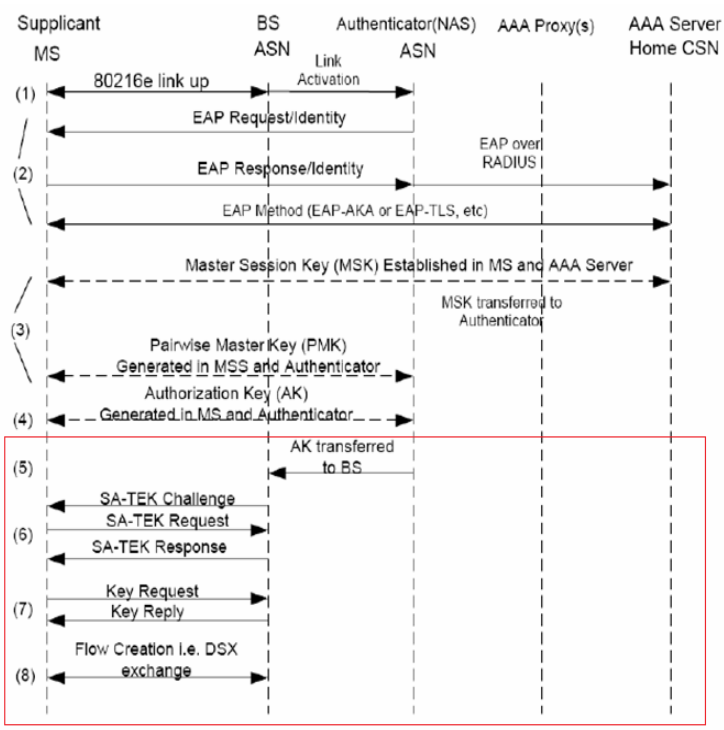

Figure 2: 802.16e-2005 authentication process. Red portion indicates the steps needed in re-authentication. Source: [5]

The last and the fastest method is the "PKD with Anticipated 4 way handshake" (PKD-A4WH). In PKD-A4WH, mobile, in initial authentication when first enters the network, establishes ALL the keys including PTK and GTK with current AP AND neighboring APs. Thus, during handover, mobile can immediately resume transmission without any key setup at all.

All these methods do not add/remove steps from the 802.11i protocol, and, hence, little security modification to the original 802.11i [9]. They only creatively move some steps from re-authentication stage in handover to the pre-authentication stage in initial connection.

2.2.2 Fast and Secure Handover in WiMAX

IEEE 802.16e standard, otherwise known as "WiMAX", is the next generation miles-long mobile wireless broadband. Unlike MANET, WiMAX is an hybrid of infrastructure and ad hoc network. Mobile nodes are attached to a base station infrastructure, but each base station can

communicate with each other using ad hoc links. Similar to MANET, WiMAX nodes are highly mobile, and therefore require an authentication protocol to accommodate that. There are currently three competing authentication protocols proposed for WiMAX – EAP-TLS, EAP with Subscriber Identity Module (EAP-SIM), and EAP with Authentication and Key Agreement (EAP-AKA) [4, 6]. This section presents an improvement over the version of EAP-TLS that is specifically tailored for 802.16e-2005 standard. Depending on which method used, there are different ways to establish the keys. However, all method follow the general steps listed in figure 2. Below is a more detailed description (based on [4, 14]):

1. Mobile contacts Authentication, Authorization and Accounting (AAA) server via Internet Service Provider (ISP). Through EAP-Identity Request/Response packets, the mobile identifies the BS, ISP, and AAA.

2. Mobile shares a secret symmetric key called "Master Key" (MK) with AAA server. Using MK and a couple nonces, mobile and AAA server establishes the symmetric key – "Master Session Key" (MSK) with the AAA.

3. MSK establishment authenticates mobile to the network. AAA server, after established MSK, alerts the ISP that the mobile is authenticated and sends to ISP the MSK to communicate with mobile. ISP will then proceed to establish a new symmetric key called "Pair-wise Master Key" (PMK) with mobile.

4. Using PMK and a couple nonces, mobile and ISP derives the symmetric key – "Authorization Key" (AK), and ISP sends AK to base station.

5. Base station uses AK to communicate with mobile in private, and establishes a new symmetric key called "Transport Encryption Key" (TEK).

6. Mobile can start transmission, and all transmission will uses TEK for encryption

During handover to another base station but still with the same ISP, mobile only has to re-establishes AK and TEK, which is very fast. However, when handover to another ISP, mobile has to re-do the entire authentication process.

An improvement over this is called "Fast and Secure handover for 802.16 based on pre-authentication" [6]. This proposed approach utilizes pre-authentication similar to the one discussed in previous section. The idea is that, when the mobile first connects to the network, the mobile would establish MSK and PMK with current ISP and neighboring ISP as well (via current ISP). During handover to another ISP, mobile only has to verify PMK and establish AK and TEK. This proposed method reduced entire re-authentication process in inter-ISP handover, and makes inter-ISP handover about as fast as intra-ISP handover.

2.2.3 Secure Ad-hoc On-demand Distance Vector

A significant security enhancement over original AODV is the Secure AODV (SAODV) [21]. SAODV is designed to be conscious about the resource-limitation of AODV mobiles, and it does not require a centralized authority. SAODV uses digital signature from public-key cryptography and one-way hash to protect the packet header fields [21]. However, SAODV does not protect packet payload [21]. Moreover, SAODV does not have a well-designed encryption and key-management system [21]. Most importantly, SAODV is designed around original AODV, and it may be difficult to modify to work with other MANET protocol.

III. Versatile and Extensible Security System

Versatile and Extensible Security System is divided into two primary modules – authentication module and encryption module. Each module is composed of improved and modified version of existing ideas and technologies. Both module work together to provide complete protection coverage on confidentiality and integrity of the network. The authentication scheme ensures that only authorized nodes can enter the network, and the encryption scheme ensures that all nodes in the network can talk securely and privately. Authentication and encryption scheme together provide privacy and security against external threats for communications. The third component – reputation scheme – can be extended in the future provides protection coverage on availability and ensures security against *internal* threats [11] (this is discussed more in section 5). In abstract, VESS is treating entire network like a country, and network nodes like citizens of the country. The authentication and encryption systems are the *military protecting against external enemies,* while the future-extendable reputation system is the *police force engaging in peacekeeping inside the country.*

3.1 Authentication Scheme

The authentication module is based primarily on public-key cryptography, digital certificate, and the concept of pre-authentication. Below is a broad overview of the entire authentication procedure (depicted in figure 3):

1. Node authenticates BEFORE entering network, and nodes obtain proof-of-authorization as a form of pre-authentication.
2. Node, upon entering the network, shows the neighbors the proof-of-authorization for the right to establish connection. Neighbor verifies the proof-of-authorization.
3. Re-authentication does not require node to go through entire authentication process again. Node simply presents the proof-of-authorization during re-authentication, and there is no need to re-authenticate node's identity again.

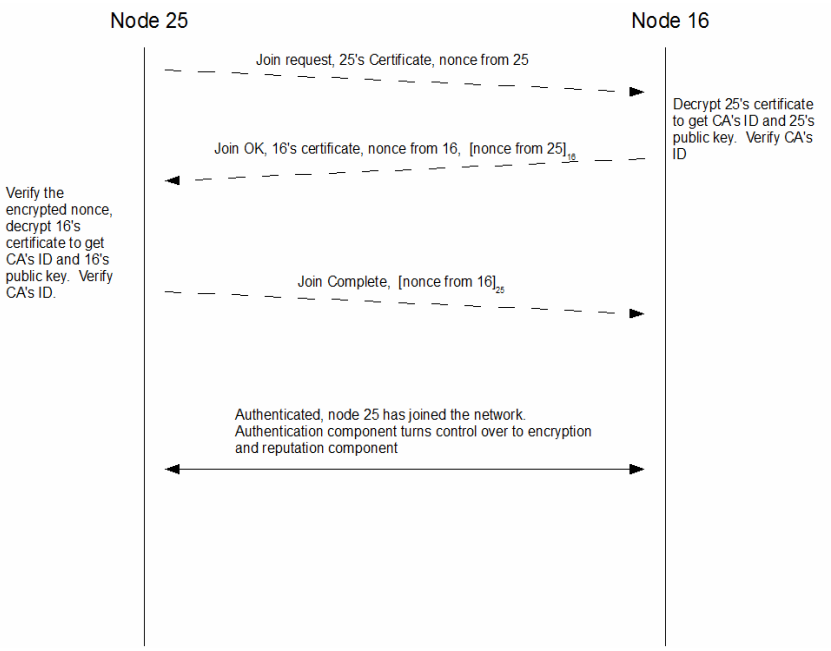

Figure 3: message exchange in authentication process

A more specific example has been included in figure 4, where it shows the step of a node 25 attempts to enter the network. In the figure, 25 first contacts CA to obtain the certificates. During entry to the network, 25 sends "Join request" packet to node 3, 16, and 27, which contain the certificate and a nonce. Node 3, 16, and 27 then sends to node 25 their respective certificates, nonce from 25 (signed with their private key), and new nonces from them. Node 25, upon receiving certificates and new nonces, reply back to each node their respective nonce (signed using 25's private key). Node 3, 16, and 27 then sends "Join Ok" to finish authentication and joining process. Authentication is placed outside the network for several advantages that will be discussed shortly. Since authentication is outside the ad hoc network, it does not matter which kind of

authentication procedure is used as long as nodes can be safely and securely authenticated and proof-of-authorization can be obtained. To make things simple, our design of VESS assumes nodes authenticate with some centralized authentication service called Central Authority (CA). This CA authenticates node via a secure and reliable mean (e.g. face-to-face, biometric, or password through a secure communication channel). This assumption of authenticating with a centralized node is realistic because nodes in mobile communication usually have to register with some kind of centralized management before they are ready to use the network [18]. Moreover, a centralized authority is much more efficient and secure than a distributed one [3, 7, 16, 18]. This fact is evident in WiMax security scheme, where a centralized Authentication, Accounting, and Authorization server exists to provide secure authentication even though base stations are largely connected in ad hoc manner [4, 6, 13, 14].

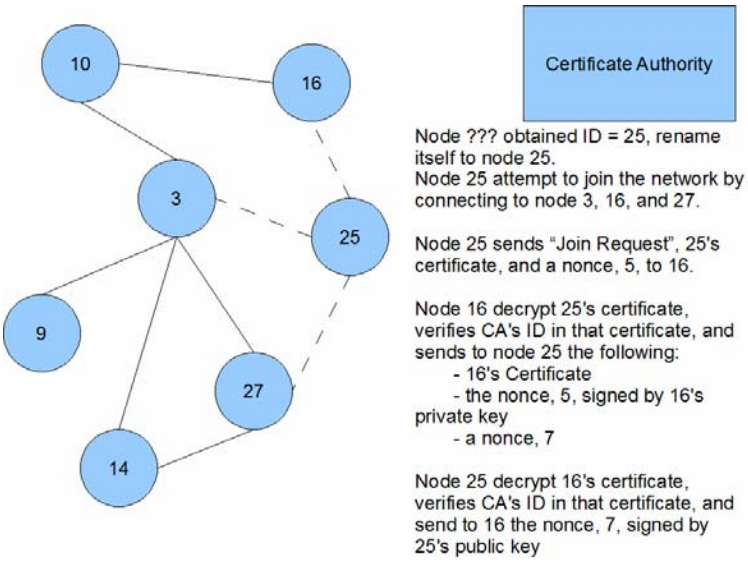

Figure 4: Network joining process, node 25 joins the network by authenticating with neighboring node 3, 16, and 27

Placing an authentication outside the network and uses only the proof-of-authorization as the ticket to enter the network has four advantages:

1. This method separates the network from majority of the authentication system, and it can help reduce attacks that are specifically targeted against authentication system.
2. Identity Authentication is now outside and independent of the security system, and thus it can be modified or upgraded in the future without affecting the security system.
3. Identity-authentication, which is the most comprehensive and performance degrading, will not affect network performance now. This is a gain for network performance.
4. Re-authentication is now faster and easier since only the proof-of-authorization needs to be verified, which is a process a lot faster and easier than identity-authentication.

Inside the network, the entire authentication process relies on the facts that proof-of-authorization is secure and nodes can securely and reliably verify the proof-of-authorization. To that end, VESS uses digital certificate based on public-key cryptography as the proof-of-authorization that nodes use to authenticate each other in the network. Digital certificate such as the X.509 certificate used by WiMax has been proven to be relatively secure and reliable [14, 18].

Initially, nodes initiate identity-authentication with CA outside the network. Once the identity of the mobile has been authenticated, the CA issues to the mobile the mobile's public-key pairs and mobile's digital certificates, which and contains the public keys of the mobile. The digital certificates and the corresponding key pairs may be pre-determined or maybe randomly generated, the CA has the choice of preference. CA usually issues multiple certificates to the mobile so that mobile can use different certificate for authentication and encryption. The certificate hash must be signed by the CA, and CA's public key can be obtained by the node during authentication. With CA's public key, nodes in the network can verify neighbors' certificates easy and fast.

For example, for a particular node N that is trying to join the network, the digital certificate is of the following layout:

(Body, $[\text{MD5 hash of body}]_{CA}$)

Body = (public key of N, N's ID, issue time, expiration time)

The bracket with a subscript "CA" indicates the hash of certificate body is sign by CA using CA's private key. This hash of certificate body can only be properly decrypted using CA's public key, and certificate verification only succeed if hash value computed by end node on the body matches the hash value decrypted from the certificate.

As soon as N joins the network, the following message exchanges take place:

- N join requests to neighbors
- Every neighbor sends to N its certificate and a nonce
- After verify the certificates using CA's public key and hash, N sends to each neighbor a message encrypted using each neighbor's public key. This message contains N's own certificate, nonce from the corresponding neighbor, and a nonce from N.

- The neighbors verify N's certificate and the nonce inside the reply message, and then reply to N an "Ok-to-join" message with the nonce from N encrypted in N's public key
- Finally, N receives all the "Ok-to-join" message, verifies the nonces, and this concludes joining procedure.

If in any step the certificate verification fails, computed hash value not matches, or if the nonce is incorrect, then the joining procedure fails and N is denied to connect to the neighbor. N may choose to try again afterward, but administrator of this security protocol may only allow a maximum of (k) number of join failure within certain amount of time (t). Both k and t are decided by network administrator.

All the certificate verification procedure must go through the following process. The protocol assumes nodes in the network have secure and reliable mean to synchronize time outside the network, the precision of time must be within +/- 1 hour from standard time in the network.

Certificate Verification Process:

1. Decrypt the hash on certificate body using CA's public key, then end node hash the certificate body itself If decrypted hash equals to computed hash, then the node goes to the next step. If two hashes are not equal, then this certificate is NOT ISSUED BY correct CA or the certificate is corrupted. In this case, certificate verification fails.
2. Verifies the expiration time of the certificate. If current time is (p) amount of time past the expiration time, then the verification fails. Otherwise, the verification succeed. Network administrator can decide the number p.

From this point on, this paper will refer to Certificate Verification Process as "decrypting certificate" and use the two terms interchangeably. After certificate verification succeed, nodes will store verified certificate in the memory. The certificate is removed from the memory when it expires or when connection to the corresponding direct neighbor is lost. Expiration time exists to keep freshness of the authentication. When certificate expires, node has to contact and authenticate with CA again to obtain new certificate. Assuming the CA is secure, the strength of authentication inside the network can be influenced by how often nodes refresh their

certificate. This is one of the place in VESS where network admin can balance between performance and security.

3.2 Encryption Scheme

VESS encryption scheme is based off distributed public cryptography, but VESS has four levels of encryption design. Users of the VESS can choose between a balance of security and performance by choosing a level of encryption best suited the needs.

Encryption component of VESS utilizes the digital certificate in authentication component. After certificate verification, nodes directly connected to each other should have the public keys of each other as the public keys are inside the certificate. Since the certificates are signed by Central Authority, no unauthorized third-party can manipulate the certificates or the keys inside them. Therefore, for nodes that are directly connected to each other, the public keys they have about each other are always authentic.

To protect key exchange between a pair of source and destination, VESS deploys the concept of certificate chaining used in distributed public cryptography [10]. The foundation belief behind certificate chaining is that nodes always have public-keys of directly connected neighbors (via certificate verification), and so nodes can vouch for the encrypted packet from directly connected neighbors and vice versa. Distant pair of nodes wishing to exchange public-key can send each other public-key in a special packet, and they can have the hops between them successively vouch for this special packet. This successive vouching ensures

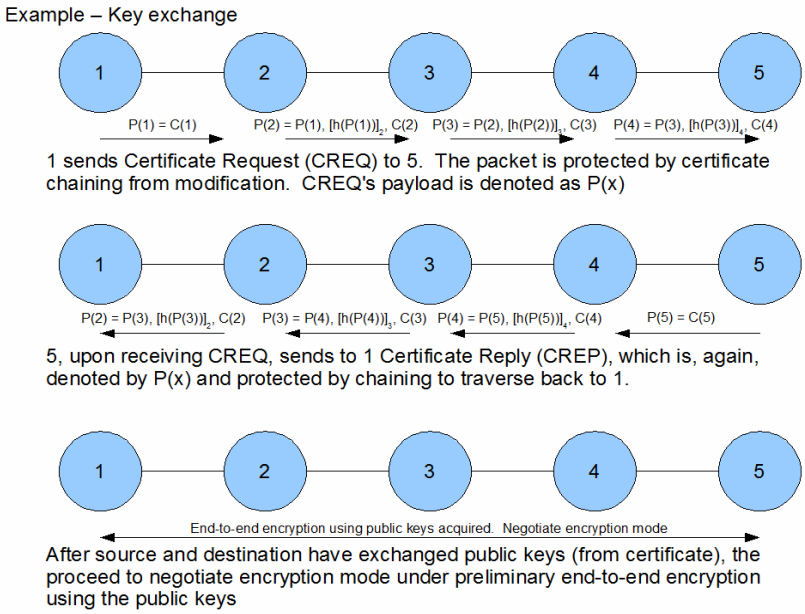

Figure 5: example of certificate chaining and key exchange

the special packet is not modified, and certificate from destination can safely and securely arrive to source via this chain of trust.

In VESS, when source wants to acquire the public-key from destination, source sends to destination a special packet called "Certificate Request" (CREQ) packet. CREQ will contain inside its payload, the certificate of source. Upon receiving CREQ, destination sends a special packet called "Certificate Reply" (CREP) back to the source. CREP contains destination's certificate in the payload. Both CREQ and CREP traverse through network via certificate chaining. We will now explain the concept of certificate chaining in detail.

The concept of certificate chaining is quite intuitive. Assuming a node sends special packet such as CREQ or CREP to an intended node. The first hop of traversal, which is the direct neighbor of the sending node, will hash the payload, and sign that hash. The first hop will then appends its own certificate and the signed hash to the original payload; this forms the new payload. Finally, it will send the new packet with the new payload to the next hop. The next hop, which is the direct neighbor of previous hop, will do the same. This successive chain of hashing, signing, and certificate adding is called certificate chaining, and this is how each hop vouches for the integrity of the special packet received from previous hop.

Figure 6 shows an example of certificate chaining. In figure 6, there is a simple route of 5 nodes connecting to each other in series (1 connects to 2, 2 connects to 3, 3 connects to 4…and so on). Let 1 be the source and 5 be the destination. In order for 1 to obtain the certificate from 5 safely, 1 sends CREQ to 2. The payload in CREQ will contain the certificate of 1. 2, upon receiving CREQ, hashes the old payload, sign the hash, append the signed hash and its certificate to old payload to form new payload, and then send the modified CREQ to 3. The rest of the hops (3, and 4) will do the same. The final packet received by 5 will contain the following payload:

*C(5) = certificate of 5
*h(x) = hash of data x
*p(3) = payload composed by 3, received by 4

$p(3), [h(p(3))]_4, C(4)$

By verifying the computed hash against the signed hash, node 5 verifies that node 4 vouches for the payload received from node 3, and then proceed breaking down p(3).

p(3) will contain these items:

p(2), $[h(p(2))]_3$, C(3)

Node 5, again, verifies the computed hash of p(2) against signed hash $[h(p(2))]_3$, and this tells node 5 that node 3 vouches for the payload received from node 2. After verification, node 5 then proceeds to break down p(2).

P(2) will contain the following items:

p(1), $[h(p(1))]_2$, C(2)

Using the same verification method, node 5 make sure node 2 vouches for the payload from node 1 and proceed breaking down p(1). p(1) will simply contain the C(1), which is the certificate of 1.

Thus, the destination, 5, by reading the certificate in each layer and then compare the signed hash value for each the sub-layer, can safely acquire the certificate of 1. In an abstract sense, this message represent a chain of trust such that 2 vouches for the authenticity of CREQ payload from 1, 3 vouches for CREQ payload from 2, and, finally, 4 vouches for CREQ payload from 3. Since 4 is the direct neighbor of 5, 5 already trusts any packet from 4, and this chain of vouching guarantee the certificate of 1, received by 5, is authentic. Upon receiving the certificate from source, destination will send CREP to source via certificate chaining.

Certificate chaining provides two essential functions:

1. Defense against compromised certificate and private key: in the event when a private key/certificate is compromised, there is a possibility for a malicious attacker to use this compromised (but not yet expired) certificate and private key to launch man-in-middle attacks. Certificate chaining allows each node to vouch for the originality of the certificate, and help prevents such attacks. For information about this can be found in section 5.1.

2. Lays framework for future extension: certificate chaining is useful for future extensions that are intended to defend against attacks on availability. For example, reputation system can use certificate chaining to help detect and locate flooding attacks.

Certificate chaining guarantees a pair of node can securely exchange public-keys, and this is the foundation of VESS encryption system. VESS provides four different levels of communication encryption for MANET. Besides nodes in Open Mode (discuss shortly), each node using the VESS will have an Encryption Table. Each entry in the Encryption Table stores encryption information established with a node. For VESS, each entry in the Encryption Table is in the format below

[Node ID, Expiration of the key, Type of encryption, encryption key]

The node ID field stores the ID of the node that uses the encryption. The encryption type field indicates one of the three types of encryption in VESS that is used to talk to the node with the corresponding node ID, the encryption key field stores the key for the encryption, and, finally, expiration field tells when the key expires.

The four level of encryption are as following:

1. Open Mode: no encryption.
2. Lightweight Mode: low encryption strength for better network performance
3. Strong Mode: heavy encryption at the expense of network performance.
4. User Mode: a mode to let user decide how to balance encryption strength and performance
5. Data-integrity Mode: a mode for data integrity check. It can be activated as a stand-alone mode, or dual-activate with another mode.

3.2.1 Open Mode

Data in this level of encryption are all send in the open without any confidentiality. This level is used when network doesn't have sensitive data that needs to be protected from public reading. Each pair of hops in packet route can sign the packet pair-wise using the known certificate from each other, and this can protect packet modification. Since nodes outside the network cannot authenticate into the network, these nodes cannot use services provided by the network even

without encryption. This level of encryption is best suited for wireless sensor network used in academia and research. An example of this network would be MOTE, the environmental monitoring sensor network from Berkeley. MOTE records environmental data for academic research purpose, and the data collected are available online for everyone. In this case, encryption would not be necessary since the data collected are already freely distributed.

3.2.2 Lightweight Mode

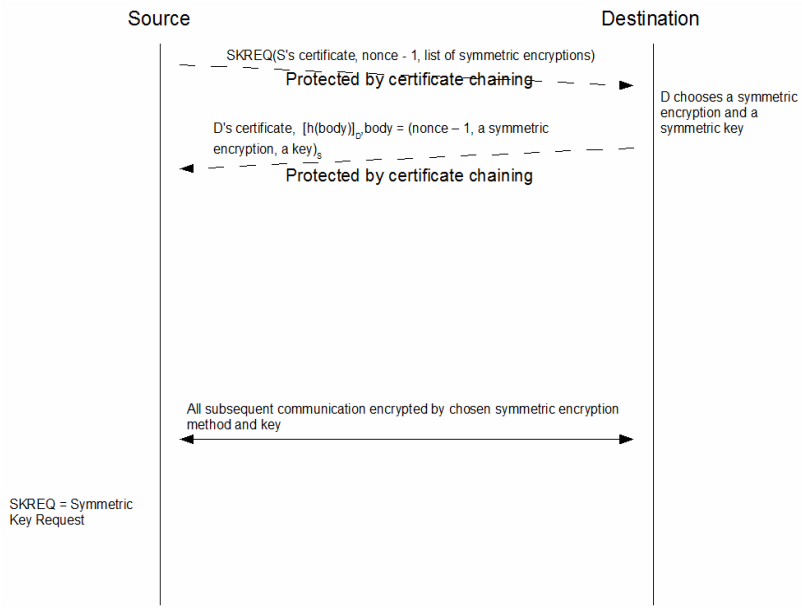

Figure 6: message exchange to establish lightweight mode encryption

This level of encryption provides a pair of nodes with better network performance at the expense of slightly weaker encryption strength. Performance gain is achieved by using smaller key length and use key exchange procedure only once. Figure 6 depicts the entire process of lightweight mode establishment. A more detailed description of steps is included in the following.

Initially, for a pair of nodes that wish to communicate in private, the source initiates Lightweight Mode encryption by sending a special packet called "Symmetric Key Request" (SKREQ) to the destination. SKREQ payload contains source's certificate C_S, a nonce chosen randomly by the source N_s, and a list of symmetric encryptions supported by the source L_S. Destination, after receiving SKREQ, randomly select another nonce N_D, and sends a "Symmetric Key Reply" (SKREP) packet to source. The SKREP would include in the payload the destination's certificate, and payload body. Payload body contains N_D, N_S, and a encryption method chose from L_S. SKREP payload body is encrypted by source's public key for security. A signed hash on payload body is also included in the payload for security. Both SKREQ and SKREP are signed and protected by certificate chaining during their traversal. After the source receives the SKREP, both source and destination would have N_D, N_S, and the public keys of both source and

destination (received during certificate chaining). With these materials, source and destination can derive a common symmetric key and all subsequent communication will be encrypted using the chosen symmetric encryption method and the derived key. The expiration time for this level of encryption is user-decided. Nodes with few resources can make the encryption key valid until their certificate expires. As long as performance permits, nodes should refresh key and encryption frequent to keep the encryption secure. When the encryption expires, node sends to its communication partner a Symmetric Key Expiration notice (SKEXP).

3.2.3 Strong Mode

Strong mode is offers the best encryption strength available in VESS. In this level of encryption, key and encryption are REQUIRED to be refreshed periodically. When a pair of nodes wish the communicate using this level of encryption, the two nodes would first exchange certificate using certificate chaining. Figure 7 depicts the steps in strong mode establishments. The following is a more detailed description of the figure. After acquiring public keys, all the subsequent key exchange packets would be encrypted using public cryptography. The pair of nodes would send each other a nonce chosen at random and negotiate a symmetric encryption method and a cryptographic hash function. Using the two nonces and both public keys, the chosen cryptographic hash function will hash a key for the chosen symmetric encryption. All data communication from then on will be encrypted using the selected symmetric encryption and key.

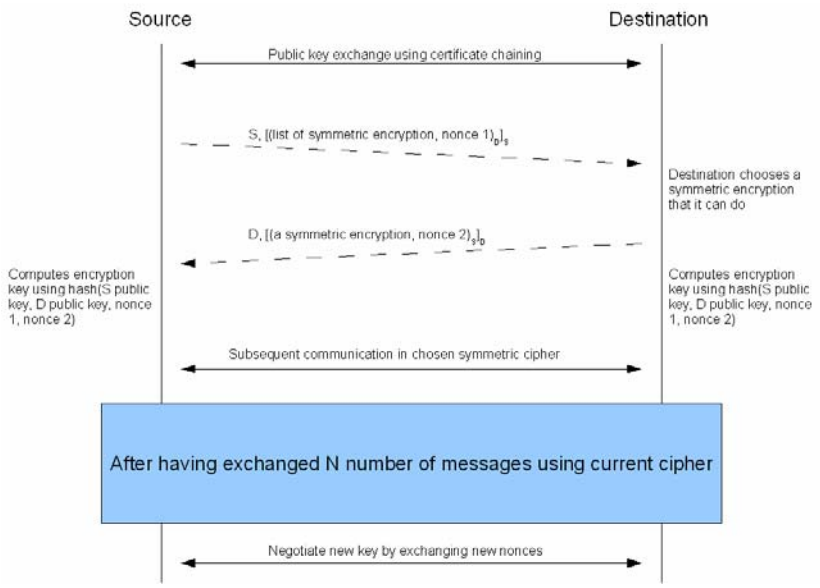

Figure 7: Message exchanges to establish strong mode encryption

To ensure the encryption strength is fresh, the encryption expires very quickly. Each symmetric encryption expires after every 200 encrypted data messages. After 200 messages, a new key and new symmetric encryption method is negotiated under current symmetric encryption.

3.2.4 User Mode

User mode is a highly customizable mode designed for user to customize their own balance of encryption and performance. Key exchange procedure in this mode is exactly the same as the procedure in Strong mode. Except in this mode, users can decide the symmetric encryption algorithm to use, the symmetric key length, and the key refresh frequency. Below is the preset limitation for each variable:

Symmetric Encryption: DES/AES

Symmetric key length: (bits)

Key Refresh Frequency: -1~65536 (seconds)

The key refreshes according to the key refresh frequency. For example, if key refresh frequency is set to 10 seconds, then the new key exchange occurs every 10 seconds. If the key exchange frequency is set to 0, then the symmetric key negotiation only occurs once. By default, User Mode uses DES, set the key refresh frequency to 0, and key length is set to 56 bits. Therefore, the default setting of User Mode is just "Lightweight Mode". When Key Refresh Frequency is set to -1, the symmetric encryption module is turned off. In this case, the pair of nodes encrypt the data using the public key that they have acquired from each other. This is an option offered to the users as part of the adjustability and versatility of VESS.

User mode does not offer automatic negotiation of key bit length. A pair of node, before start using User Mode, must agree upon the key length and refresh rate. For example, a pair of node can initiate Lightweight Mode first, and then communicate under Lightweight Mode to discuss the key length and refresh rate. User has to negotiate and set the key length and refresh rate manually, and hence the name "User mode"

3.2.5 Data-integrity mode

Data integrity check, a vital component of information security, is not included with all the previous mode. To patch up this flaw, data-integrity mode is introduced. Data integrity mode utilizes digital signing and cryptographic hash function to automatically check on packet tampering and corruption. Data-integrity mode can be activated as a standalone mode or it can be dual-started with one other mode to add automatic data integrity check to this one other mode.

During initialization, data-integrity module first checks if there is another mode to dual-start with. If there is a dual-start mode, then there is nothing to do in the initialization phase. If there is no dual-start mode, then data-integrity module starts certificate exchange between source and destination via certificate chaining.

After initialization phase is completed, data-integrity module quietly wait for data arrival. In the protocol stack, data-integrity mode sits in between network layer and other security modes. Data-integrity module automatically encapsulates payloads from other security modes or transport layer, and create a new payload for network layer. For the data-integrity module of a particular node, N, the module creates a format like below:

$([\text{hash of payload}]_N$, payload from transport layer or other security module)

For data that arrives from network layer, data-integrity module first decrypt the signed hash and then compute the hash of payload. If the decrypted hash and computed hash matches, then data integrity check pass (integrity check fails otherwise).

For data that arrives from transport layer or other security module, data-integrity module hashes the payload, and then sign the hash. The signed hash and payload are encapsulated in the format described above, and then data-integrity module forwards this encapsulate packet to network layer.

Table 1: Comparison of encryption mode

Mode	Open	Lightweight	Strong	User Mode at -1
Main Encryption Type	No encryption	Shared-key cryptography	Shared-key cryptography	Public-key cryptography
Main Encryption Standard	N/A	56 Bits DES	56 Bits DES/AES	1024 Bits RSA
Key Freshness	N/A	Set to 0	Key update every 200 messages	Set to -1
Encryption Strength	NONE	Normal	Strongest	Strong
Memory Complexity (for encryption only)	NONE	N (2N if dual-start)	3N	3N
Message Complexity (for encryption only)	NONE	2N	4N	4N
Time Complexity (for encryption only)	N/A	Small	Medium	Big

3.3 Security Analysis

The security strength of VESS is analyzed in this section. There are infinite possible attacks on security, and it is impractical to try to simulate or discuss them all [1]. Furthermore, there may be some unforeseen attacks in the future. For this reason, the report only presents how current VESS design can ensure security, and VESS is left open for future modification and extension.

3.3.1 Authentication

Digital certificate in VESS is based on public cryptography. The security of certificate itself depends entirely on the security of the public cryptography used and the security of CA. Assuming CA and the utilized public cryptography is secure, and CA always authenticates nodes

with no error or security breaches. Neighbor's pair-wise certificate authentication should have no security weakness.

In the case of certificate exchange using certificate chaining, security can be breached by having a malicious node modifying the exchange packet. However, such an attack could not be carry out without affecting the signature on the exchange packet. Thus, such a modified packet would be detected immediately, but there will be no way to detect and eliminate the malicious node causing the attack. To detect and eliminate malicious node, a reputation system has to be included with VESS. This is discussed more in section 5.

3.3.2 Open Mode

This level of security offers only authentication protection, and absolutely no encryption. Therefore, eavesdropping, packet insertion, packet modification, and packet forging attacks are all probable. This level of security is developed for those MANETs that malicious attackers find unworthy of the effort to attack.

3.3.3 Lightweight Mode

This level of security is designed for MANET that does not requires very strong encryption but values network performance. For example, an emergency communication network in the event of natural disaster, multimedia streaming, or gaming network…etc. Since symmetric key is not refreshed (not before certificate expires), it is possible to break the encryption using brute force attack for a powerful malicious entity with rich resources (e.g military). Although cryptanalysis is highly unlikely to succeed if a good encryption standard (such as the DES in our implementation) is used. However, how the destination chooses the key can influence the strength of encryption. The encryption is considered weak if destination uses the same key for an extended period of time.

3.3.4 Strong Mode

This level of security has the best encryption strength VESS can offer. The public-key cryptography from the certificate is used at minimum to reduce the number of messages that malicious entity can use in cryptanalysis. The symmetric encryption in this level is forced to be refreshed every 200 messages. This forced periodically refresh not only makes sure malicious

entity cannot obtain data for cryptanalysis, it also keep up the strength of encryption by changing to a new encryption type and key before current ones are broken. Strong mode also offers nodes perfect forward secrecy if used correctly. To achieve perfect forward secrecy, nodes can discard nonce as soon as key is computed, or replace nonce with timestamp. Backward secrecy depends entirely on the security of CA, however. If this encryption level is used correctly as intended and the chosen symmetric and public-key encryptions are secure, then breaking of encryption level is infeasible if not impossible.

3.3.5 User Mode

This level of security is recommended for MANETs users who wish to customize their own balance of encryption strength and performance. Even at the default setting, which is equivalent to Lightweight mode, the encryption is already secure. However, like the situation in Lightweight Mode, symmetric encryption in this mode can be broken if key length is too short or key is not refreshed often enough.

User can increase encryption strength by increasing key length or using shorter key refresh frequency. Developers can also add more encryption algorithms in the future. These are vital parts of the adjustability, versatility, and extensibility of VESS. User Mode is recommended for developers, security/IT experts, or people who have knowledge on information security. The authors highly recommend ordinary users to consult with the mentioned experts before start using User Mode.

3.3.6 Data-integrity Mode

Data-integrity mode, by itself, protects the integrity of data packet and detects malicious tampering or corruption of the data packets. Since one-way cryptographic hash function and digital signature are used, it is computationally infeasible to break data integrity check if implemented and used properly.

IV. Performance Evaluation

Simulation is used to observe and evaluate VESS overhead on network and how it affects network performance in term of delays and throughput. For the simulation, the Java-based multi-purpose simulator, J-SIM, and its network components are used [20]. To observe the performance impacts and differences, eighty variants of simulations were designed. These eighty variants are as following:

- Native AODV with no VESS or other security system running
- AODV with VESS Open Mode encryption level
- AODV with VESS Lightweight Mode encryption level
- AODV with VESS Strong Mode encryption level
- AODV with VESS User Mode with refresh rate at -1
- AODV with VESS Data-integrity mode
- AODV with VESS Lightweight Mode dual-start with Data-integrity Mode
- AODV with VESS Strong Mode dual-start with Data-integrity Mode

Each variant is simulated under different traffic model, and throughput and delays in each simulation run were recorded for comparison. The traffic model tested in simulation are as following:

- Constant Bit Rate (CBR): this model simulates media streaming, remote operation, and file transfer..etc.
- Exponential On/Off: this model simulates Voice over Internet Protocol and gaming
- Poisson: this model represents normal data traffic such as web browsing and e-mail.

The authentication part of VESS, in the simulation, uses digital certificate based on the 1024 bits RSA encryption that is included in the Java Security Package. While Open Mode has only authentication and nothing else, lightweight mode uses a 56 bits DES encryption for the symmetric encryption portion. Since User Mode has refresh rate sets to -1, this mode uses the public-key cryptography that comes with the authentication; the 1024 bits RSA encryption. Strong mode use the 1024 bits RSA for the initial key exchange, but 56 bits DES or AES algorithm are used to simulate the fast refreshing symmetric cryptography in data

communication. For Data-integrity Mode, MD5 hash algorithm is used to hash data, and 1024 bits RSA is, again, used to sign the hash.

The simulation is designed to have random mobility for each node in a simulated area of 4 square km, and there are 50 nodes in the AODV network at any time. Nodes all use 802.11 for Link-layer, and each node has up to 1 Mbps duplex wireless rate. Packet size is set to 1Kbyte. Each simulation runs simulates at least four UDP flows, and the traffic load varies from 128 kbps to up to 512 kbps, Please note that the nodes support only up to 1Mbps wireless rate, traffic load 256 kbps or above saturates the network. Network is saturated in simulation to observe VESS overhead under extreme network stress. The four encryption-only modes (Open Mode, Lightweight Mode, Strong Mode, and User Mode) are simulated in standalone environment without dual-starting with Data-integrity Mode. A special simulation on Data-integrity Mode standalone case and dual-start case are discussed at the end of this section.

4.1 Throughput

The throughput measured in the simulation reflect positively for VESS performance. Figure 6 shows the throughput in kilobits per second. This figure shows VESS performance for all encryption modes (Native, Open Mode, Lightweight Mode, Strong Mode, and User Mode with refresh rate at -1).

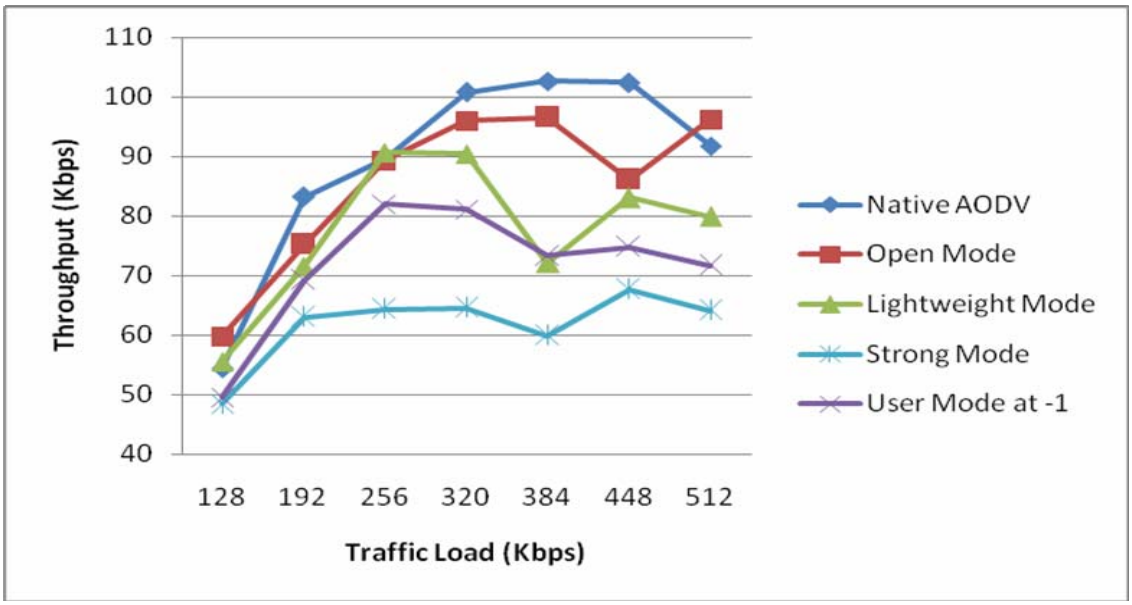

Figure 6: Exponential On/Off Total Throughput

4.1.2 Throughput – Authentication Effect

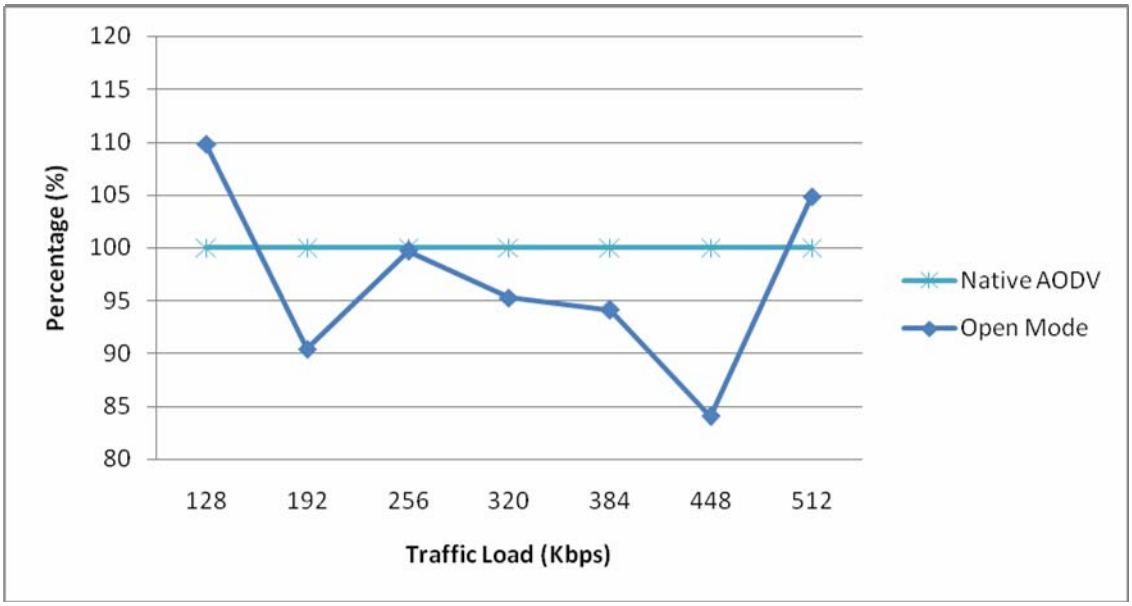

Figure 7: Exponential On/Off Authentication Throughput Ratio – Open Mode vs. Native AODV

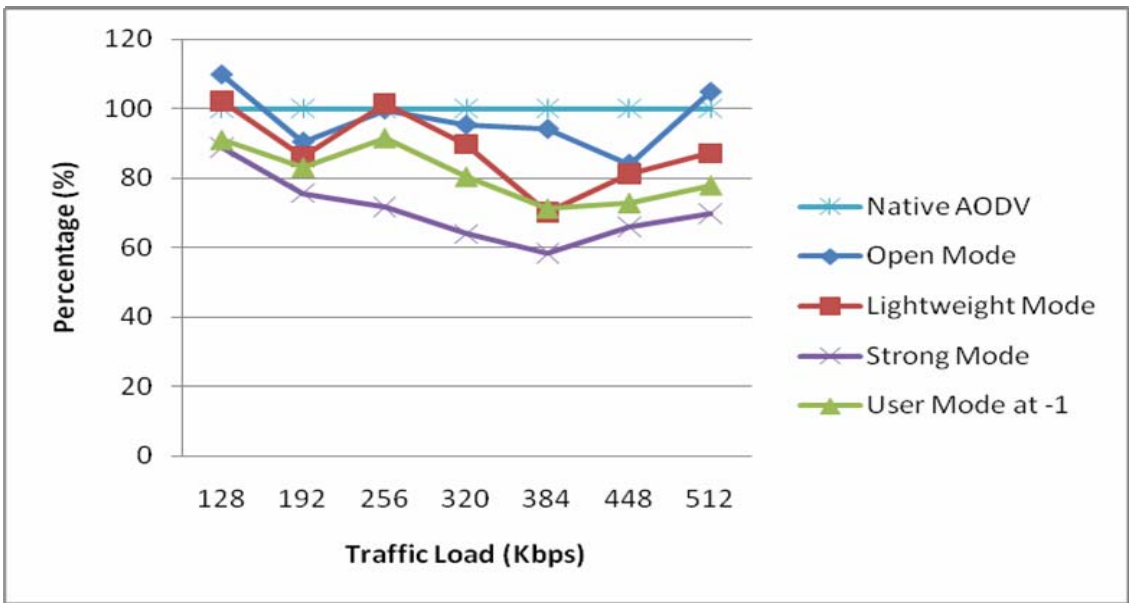

Figure 8: Exponential On/Off Throughput Ratio vs. Native AODV

Figure 7 show that the authentication module alone achieves an average of around 95%~100% of native AODV throughput. Even on worst case scenario at highly saturated network (such as the data on 448 Kbps traffic load), the authentication module can achieve an impressive 85% performance. Figure 8 shows that, even the most computation-intensive mode, Strong mode, can achieve around 60% performance of Native AODV.

4.1.3 Throughput – Encryption Effect

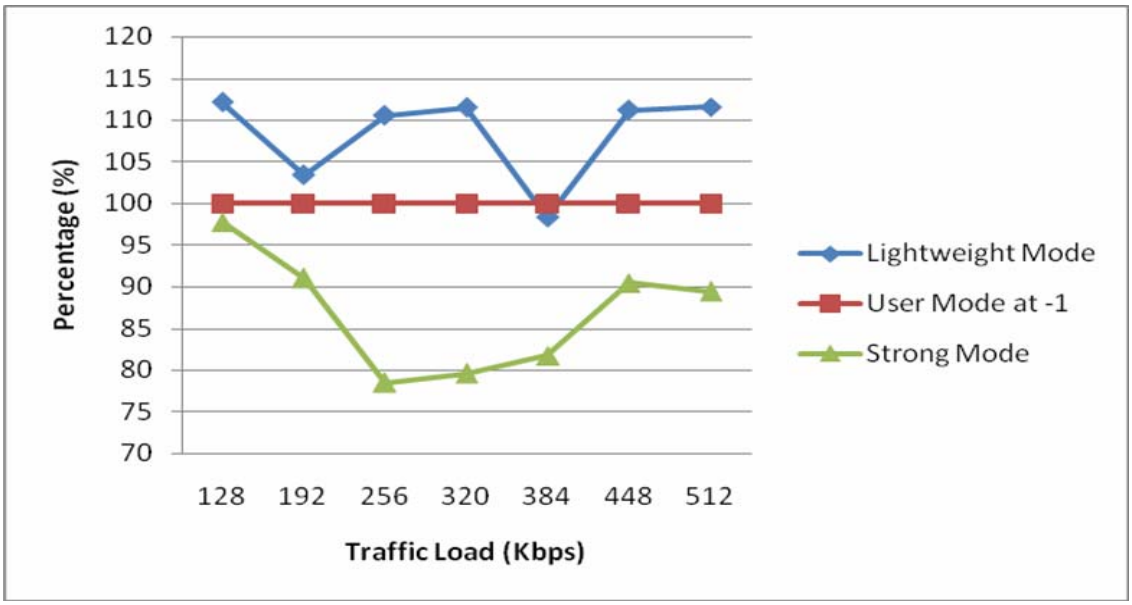

Figure 9: Exponential On/Off Encryption Throughput Ratio – Lightweight and Strong modes vs. User Mode with refresh rate at -1

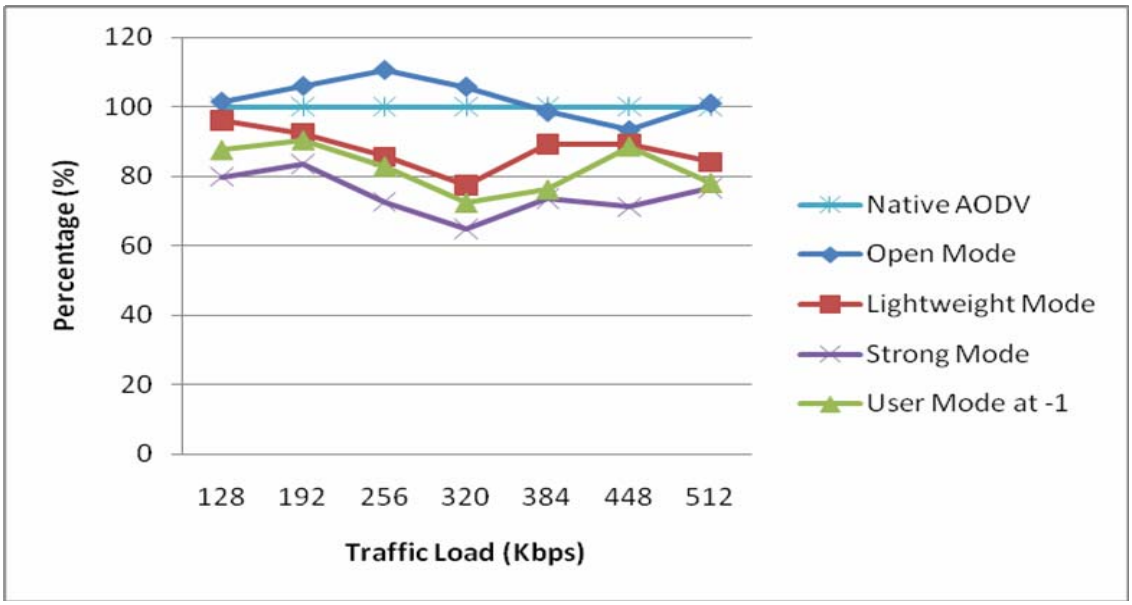

Figure 10: Constant Bit Rate Throughput Ratio vs. Native AODV

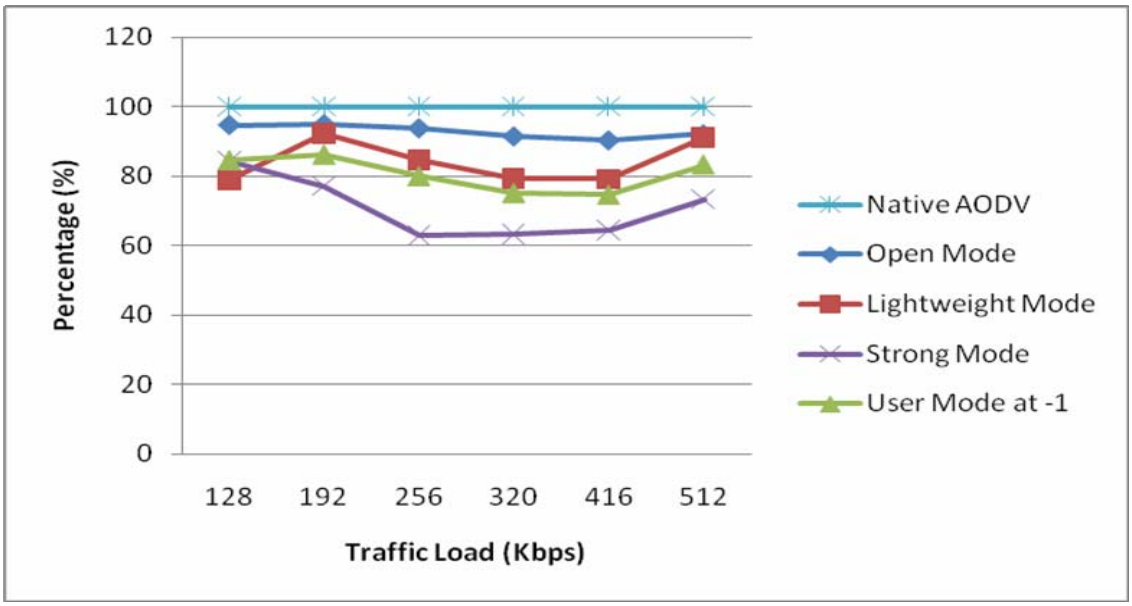

Figure 11: Poisson Throughput Ratio vs. Native AODV

Figure 9 is a graph normalized to User Mode with refresh rate at -1. We use User Mode to benchmark the Lightweight Mode and Strong Mode so that users of User mode can get a general

sense of performance vs. encryption strength. This data graph shows that, compare to the said User Mode, lightweight mode achieve on average about 10%~15% BETTER performance. Strong mode can achieve an average of 85%~90% of the performance achievable. This justifies the existence of lightweight mode and Strong mode. Lightweight achieves enough performance to justify the lessen encryption while the performance sacrifice of Strong Mode is acceptable considering the increased encryption. Finally, this graph also shows the performance relative to the key refresh rate.

Figure 10 and Figure 11 prove that, in each traffic model, VESS performance achieves a as high as an average of 95% to native AODV performance. Even the most costly encryption – Strong Mode, manage to achieve more than 60% of native AODV performance. Open Mode usually performs around 95% performance of native AODV while lightweight mode follows closely with only an average of 5% performance difference from native AODV. User Mode at -1 achieves an average of 85% performance, which is a surprisingly good performance considering this mode is using public-key to encrypt data.

4.2 Delay

While VESS throughput measurement is promising, packet delays are measured to analyze the latency caused by security overheads. Since the simulations, based on J-SIM, support only 1 Mbps wireless rate for nodes, the packet transmission time in the simulation is relatively high when comparing to networks using technologies such as 56 Mbps (802.11g) or 150 Mbps (802.11n). Hence, long delays, especially in high traffic loads, were observed in the simulation. Nevertheless, the objective here is to compare the delays caused by VESS to the delays incurred normally by native AODV. This objective is not affected by the overall long delays.

4.2.1 Delay – Authentication Effect

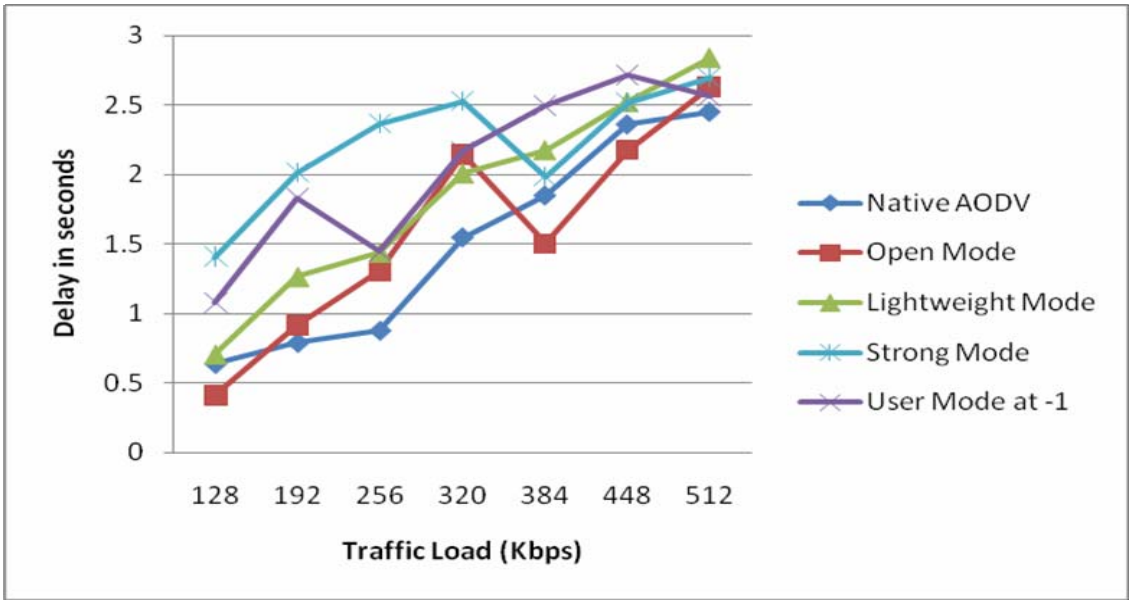

Figure 12: Exponential On/Off Total Delay

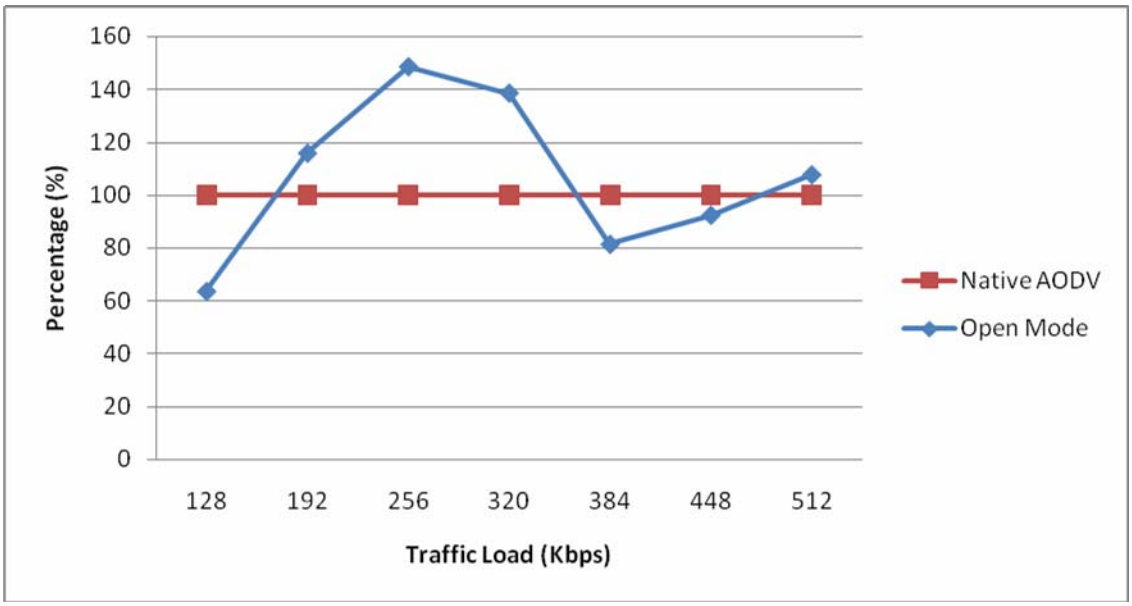

Figure 13: Exponential On/Off Authentication Delay Ratio – Open Mode vs. Native AODV

Although figure 12 shows an average of 1 second more delay for VESS at exponential on/off traffic model, VESS actually performance reasonably well during high traffic load. This is because security overhead is more apparent when there is no delay from traffic overload. When the network is saturated, the security delay becomes insignificant when compare to the delays from vast amount of data transmission.

Figure 13 show that, at exponential on/off traffic model, authentication module alone causes an average of 20% more delays. This may sound bad on the surface, but 20% actually is not much when average delays of native AODV is low. In a network with 56 mbps wireless rate, 20% of an average of 0.3 seconds native AODV delay is only 0.06 second. 0.06 second more delay is highly tolerable.

4.2.2 Delay – Encryption Effect

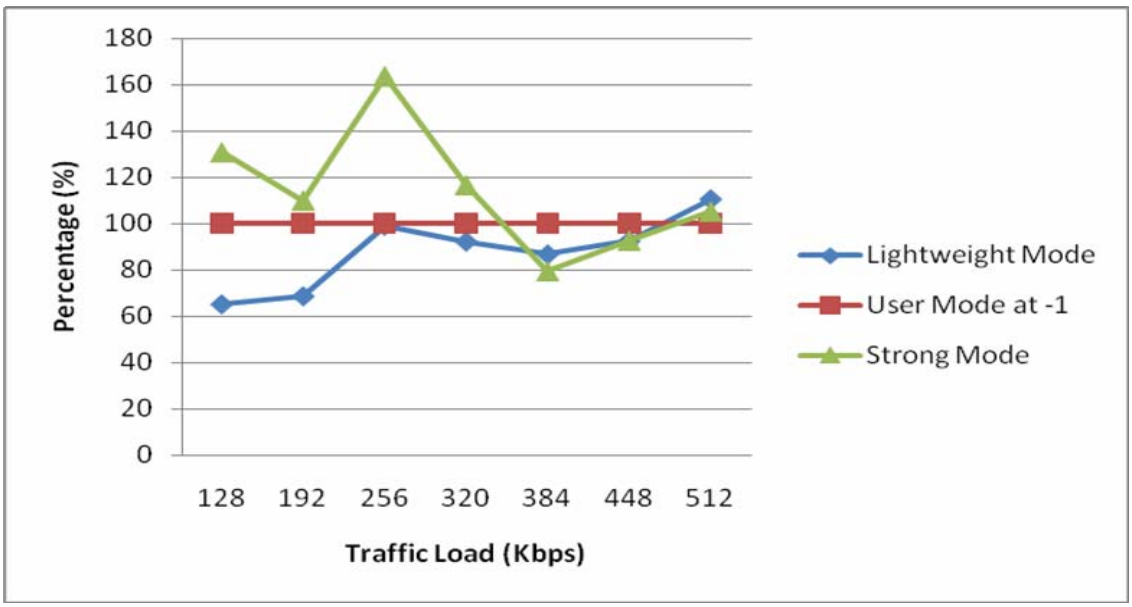

Figure 14: Exponential On/Off Encryption Delay Ratio – Lightweight and Strong modes vs. User Mode at -1

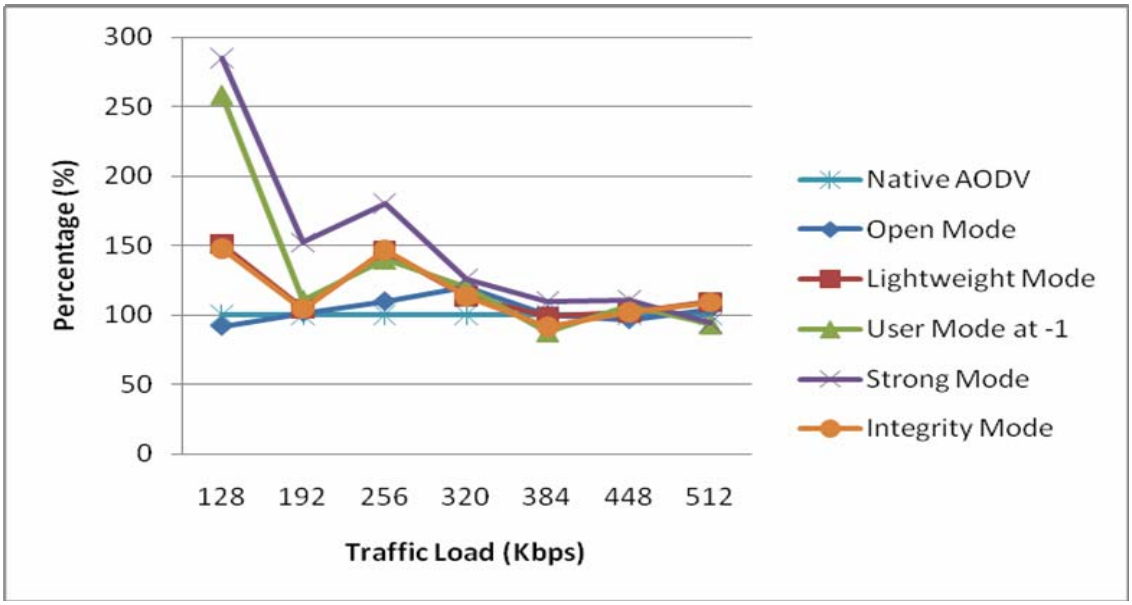

Figure 15: Constant Bit Total Rate Delay Ratio

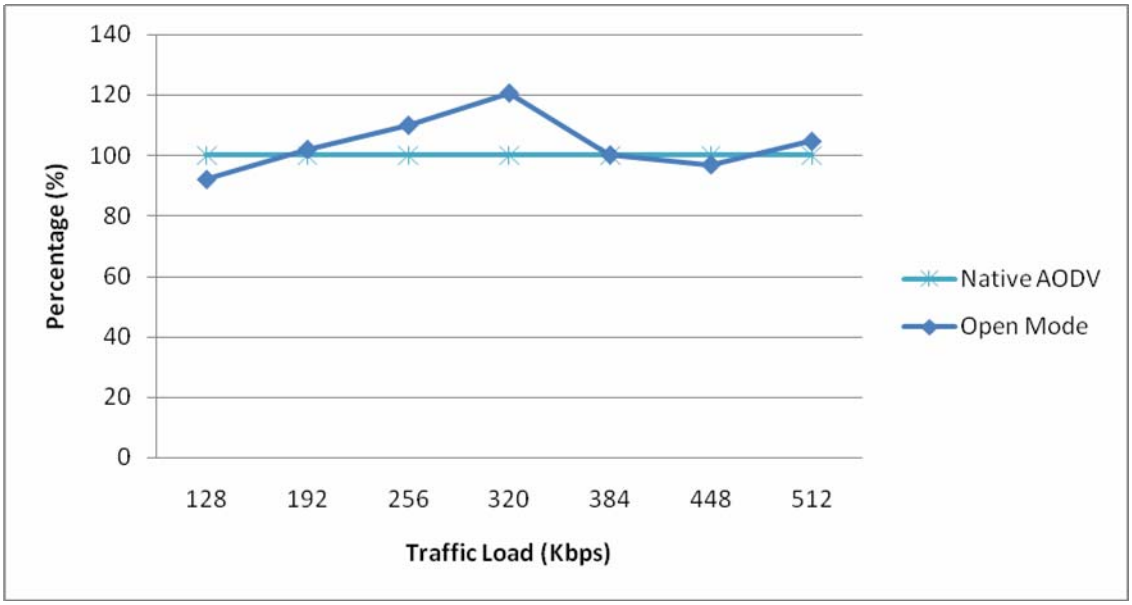

Figure 16: Constant Bit Rate Authentication Delay Ratio – Open Mode vs. Native AODV

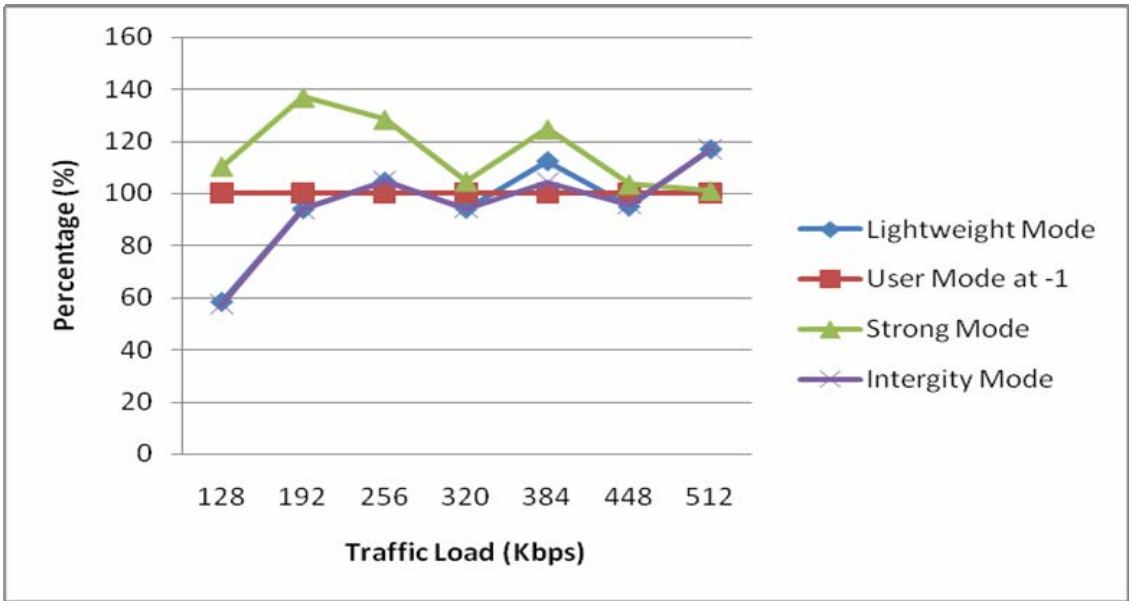

Figure 17: Constant Bit Rate Encryption Delay Ratio – Lightweight and Strong modes vs. User Mode at -1

Figure 14 serves as a proof for the justification on the existence of strong mode and lightweight mode. At exponential on/off traffic model, lightweight mode achieves an average of 20% less delays than User Mode at -1, which shows the acceptable performance boost in exchange for lessen security. Strong mode, while secure and powerful, still manage achieve an average of no more than 40% more delays than User Mode at -1. This performance degradation is tolerable considering the added strength of the encryption.

Figure 15, 16, and 17 shows that the delay result of simulation running in constant bit rate traffic model, it serves as proof that the results are comparable even in other traffic model. The high delays at 128 bit traffic load is a rare outlier caused by random node mobility.

4.3 Data-integrity Mode Performance

Figure 15 and figure 17 shows that, Data-integrity mode, by itself, performs closely to the performance of lightweight mode. Data-integrity mode performs worse than Open Mode due to the hashing and hash signing of the packets. Although not shown in figure here, when dual-started with Lightweight Mode or Strong Mode, Data-integrity Mode only adds less than 1% more delay or throughput impact to the original performance of the said two encryption modes.

V. Conclusion

5.1 Future work and possible extension

As mentioned in previous sections, VESS currently has no defense against internal Denial of Services (DoS) attacks and compromised nodes. However, future modification and extension to current VESS is easy and fast. Current framework of VESS provides many useful tools to add more security features. For example, the certificate chaining system can be used to detect possible flooding and identify the source of any man-in-middle attacks. This may be done by identifying the head section of the certificate chain. Compromised node can also be resisted by occasionally requiring data packets to go through certificate chaining. VESS also lays a strong foundation for a reputation system [5, 12, 15], which is a future module of VESS that covers the security domain of internal defense and availability.
In a nutshell, the reputation system will be a passive monitoring system that identify man-in-middle attack and check against many internal and availability attacks [2]. Packet dropping can be detected when a neighboring node, within radio range, fails to forward a packet within certain time. The nodes do not actively monitor packet forwarding, instead, nodes only monitor the next hop of a packet when a packet comes in.

For instance, when a node A receives a packet for forwarding, node A forward the packet to the packet's next hop – node B. After the packet has been forwarded to B, A monitors B to see if B continues to forward the packet to the hop after B (if B is not the destination).

To detect packet modification, nodes send special probe through certificate chaining or request data packets to go through certificate chaining. For example, when a destination, B, detects numerous data corruption, B would launches a probe back to source via certificate chaining. If a certain hop, C, signs the probe but the payload inside is corrupted, then the hop before C, in the direction from destination to source, is suspected of modifying packets.

5.2 Concluding Remarks

VESS is a versatile, adjustable, and extensible security suite for AODV and other distance-vector-based routing on MANET. It requires pre-authentication with outside authentication

authority, and it utilizes digital certificate as proof-of-authorization used in internal network authentication and re-authentication. This method places authentication duty mostly at CA, and boosted re-authentication due to mobility. Offering 4 different encryption strength level, VESS is highly adjustable to fit user needs. This report presented security analysis to explain security strength and design reason. VESS performance impact have also been carefully evaluated and discussed in this report. The result shows that even the strongest security can achieve an average of 60-80% throughput while moderately increasing delay by an average of 10%~30%. This delay increase is insignificant when VESS is used in newer 150Mbps wireless technology such as 802.11n, as the security delay does not increase linearly. Please also note that current implementation of VESS, in the simulation, is not optimized. In fact, this simulation presents the worst-case possible performance for VESS. In an optimized implementation, VESS performance should increase significantly. Finally, I believe performance of current VESS design is already acceptable, considering VESS provides extensive security coverage.

VI. Reference

[1] Andel, Todd R, Yasinsac, Alec. "Surveying Security Analysis Techniques in MANET Routing Protocols". *IEEE Communications Survey 2007. 9* (4) pp. 70-85

[2] Carruthers, R. Nikolaidis, I. "Certain limitations of reputation-based schemes in mobile environments" *Proceedings of the 8th ACM Int. Symposium on Modeling, Analysis, and Simulation of wireless and mobile systems, 2005. 8* (2005) pp. 2-11

[3] Dressler, F. "Reliable and semi-reliable communication with authentication in mobile ad hoc networks." *Proceedings of IEEE International Conference on Mobile Adhc and Sensor Systems. '05. 7-10 Nov.* (1) pp. 5-6

[4] Hu, R.Q.; Paranchych, D; Mo-Han Fong; Geng Wu. "On the evolution of handoff management and network architecture in WiMAX". *IEEE Mobile WiMAX Symp. 2007. 25-29,* pp. 144-149

[5] Hu, J. Burmester, M. "LARS: a locally aware reputation system for mobile ad hoc networks." *44thAannual Southeast Regional ACM Conference 2006. 44* (2006) pp. 119-123

[6] Hung-Min, Sun; Yue-Hsun, Lin; Shuai-Min Chen; Yi-Chung Shen. "Secure and fast handover scheme based on pre-authentication method for 802.16/WiMAX infrastructure networks". *TENCON 2007, 10th IEEE Regional Conference. 30 Oct – 2 Nov* (2007) pp. 1-4

[7] Kargl, F.; Schlott, S.; Kenk, A.; Geiss, A. "Securing ad hoc routing protocols" *30th Euromicro Conference '04.* pp. 514-519.

[8] Kassab, M; Belghith, A; Bonnin, J. "Fast and secure handover in WLANs: An evaluation of the signaling overhead". *IEEE Conf. on Consumer Communications, Jan. 2008.*

[9] Kassab, M; Belghith, A; Bonnin, J. and Sassi, S. "Fast pre-authentication based on proactive key distribution for 802.11 infrastructure networks" *1st ACM Workshop on Wireless Multimedia Networking and Perf. Modeling, pp 46-53, 2005.*

[10] Kitada, Y; Takemori, K; Watanabe, A; Sasase, I. "On Demand Distributed Public Key Management without Considering Routing Tables for Wireless Ad hoc Networks" *6th Asia-Pacific Symposium on Information and Telecommunication Technologies, 2005. 9-10* (2005) pp.375-380

[11] Laniepce, S. Demerjian, J. Mokhtari, A. "Cooperation monitoring issues in ad hoc networks" *Proceedings of the 2006 international conference on wireless communications and mobile computing. 3* (2006) 695-700

[12] Li, J., Moh, T.-S., and Moh, M., "Path-Based Reputation System for MANET Routing," accepted to present at *the 7th International Conf. on Wired / Wireless Internet Communications (WWIC),* to be held in Enschede, the Netherlands, May 2009.

[13] Lopez, R.M.; Perez, G.M.; Gomez Skarmeta, A.F.; "Implementing RADIUS and diameter AAA systems in IPv6-based scenarios" *19th International Conference on Advanced Information Networking and Applications, 2005. 2*
(28~30 March) pp.851-855

[14] Mandin, Jeff. "802.16e Privacy Key Management (PKM) version 2" IEEE 802.16 Broadband Wireless Access Working Group Project. http://ieee802.org/16

[15] Moh, M. and J. Li. "A survey of reputation and trust systems for mobile ad-hoc network routing." Accepted to appear in *Handbook of Communication and Information Security*, M. Stamp, ed., Springer, to be published in 2009.

[16] Ngai, E.C.H; Lyu, M.R. "An Authentication Service Based on Trust and Clustering in Wireless Ad Hoc Networks: Description and Security Evaluation" *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing. 1* (5-7, 6, 2006) pp. 94-103

[17] Pi Jian-yong; Liu Xin-song; Wu Ai; Liu Dan. A Novel Cryptography for Ad hoc Network Security. *Proceedings of International Conference on Communications, Circuits and Systems '06. 3* (25-28) pp. 1448-1452

[18] Safdar, Ghazanfar A.; McGrath, Clare; McLoone, Maire. "Existing Wireless Networks Security Mechanisms and their Limitations for Ad Hoc Networks" *Irish Signals and Systems Conference, 2006. (28-30)* 6 pp. 197-202

[19] Stamp, M. and Low, R. (2007). *Applied Cryptanalysis: Breaking Ciphers in the Real World.* Wiley-IEEE Press.

[20] Tyan, Hung-Ying. Design, "Realization and Evaluation of a Component-Based Compositional Software Architecture for Network Simulation". *2002*

[21] Zapata, Manel Guerrero, "Secure Ad hoc On-Demand Distance Vector Routing" *Mobile Computing and Communication Review. 6* (3) pp. 106-107

Appendix A: Network Simulators

During the course of this project, several software simulators were explored and tested. Although J-SIM is the simulator ultimately used in this project, there are many other great simulators that can easily rival J-SIM in simulations on networking and other fields. In this section, a list of explored simulators are discussed.

Network Simulator 2

Network Simulator 2 (NS2) has been the leading network simulator for over ten years. This C/C++ based simulator uses TCL for the scripting language, and this simulator is highly modular. Users of this simulator can easily modify/remove any existing modules, and add modules that are written in another programming language. Furthermore, NS2 offers a wide variety of supports from Computer Science community. However, due to its base-language being C/C++, building simulations using NS2 can be easy but slow. Although Computer Science and Open community extensively supports NS2, desired supports are difficult to locate. Moreover, supports for fields other than networking are almost non-existence.

Java Simulator

Java Simulator (J-SIM) is one of the newer software simulator designed for general-purpose simulation. J-SIM was designed on Java language and structured in the same way as NS2; using TCL as scripting language and allows addition of modules written in other language. J-SIM network module is well-designed and its supports, though not abundant, are easy to locate. Furthermore, J-SIM, using Sun Microsystem Java, has access to the vast Java Library. This not only adds overwhelming supports for fields other than networking (e.g. database and security), it also allows much faster and more rapid coding than using NS2. J-SIM was designed as the Java-version replacement for NS2. However, due to lack of community supports, J-SIM is far from capable in replacing the role of NS2. J-SIM does not support the wide variety of network

protocols that NS2 supports (e.g. IEEE 802.16e), and its simulation analysis tools are also much more limited than the similar tools offered in NS2.

Java Network Simulator

This now-abandoned simulation software was, originally, also intended as a Java version of network simulator. Unlike J-SIM, which was designed to be similar to NS2, Java Network Simulator (JNS) was designed and written completely from scratch. However, the authors of this software abandoned it due to low popularity. JNS is left unfinished, and it only supports a few network protocols and analysis tools. JNS can be used for rapid simulation in small-scale networks**,** but it is next to useless in general network simulation.