College of Technology

College of Technology Masters Theses

Purdue Libraries *Year* 2010

Securing Wireless Communication Against Dictionary Attacks Without Using PKI

Sarath Geethakumar
Purdue University - Main Campus, sgeethak@purdue.edu

This paper is posted at Purdue e-Pubs.
http://docs.lib.purdue.edu/techmasters/19

Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Sarath Geethakumar

Entitled Securing Wireless Communication against Dictionary Attacks Without Using PKI

For the degree of Master of Science

Is approved by the final examining committee:

Prof. Samuel S Wagstaff Jr
Chair

Prof. Melissa Dark Co-Chair

Prof. Eugene Spafford

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Prof. Samuel S Wagstaff Jr.

Prof. Melissa Dark

Approved by: Prof. Eugene Spafford 4/14/2010
Head of the Graduate Program Date

Graduate School Form 20
(Revised 1/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

Securing Wireless Communication Against Dictionary Attacks Without Using PKI

For the degree of Master of Science

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Teaching, Research, and Outreach Policy on Research Misconduct (VIII.3.1)*, October 1, 2008.*

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Sarath Geethakumar

Printed Name and Signature of Candidate

04/21/2010

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/viii_3_1.html

SECURING WIRELESS COMMUNICATION AGAINST DICTIONARY ATTACKS WITHOUT USING PKI

A Thesis

Submitted to the Faculty

of

Purdue University

by

Sarath Geethakumar

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2010

Purdue University

West Lafayette, Indiana

ACKNOWLEDGEMENT

I would like to thank Prof. Samuel S. Wagstaff Jr., Prof. Melissa Dark and Prof. Eugene Spafford of Purdue University for their guidance and feedback that helped realization of this research work.

I would also like to thank Preeti, Ashrith and Mithun for their continuous encouragement and support that kept me going and inspired me to complete this research.

TABLE OF CONTENTS

Page

LIST OF FIGURES

GLOSSARY

- AP Access Point
- EAP Extensible Authentication Protocol
- EAPOL EAP over LAN
- GTK Group Transient Key
- HMAC Hash-based Message Authentication Code
- KCK Key confirmation Key
- KEK Key Encryption Key
- MIC Message Integrity Check
- MIC Message Integrity Check
- PMK Pairwise Master Key
- PSK Pre-Shared Key
- PTK Pairwise Transient Key
- RxK Key Used to compute MIC on unicast data packets transmitted by the station
- SSID Service Set Identifier
- STA Client Station
- TK Temporal Key
- TKIP Temporal Key Integrity Protocol
- TxK Key Used to compute MIC on unicast data packets transmitted by the AP
- WPA Wi-Fi Protected Access

ABSTRACT

Geethakumar, Sarath. M.S., Purdue University, May, 2010. Securing Wireless Communication Against Dictionary Attacks Without Using PKI. Major Professors: Samuel S. Wagstaff Jr. and Melissa Dark.

Security of 802.11x wireless encryption standards are increasingly coming under scrutiny as compared to other security protocols and standards. The attacks on 802.11x wireless security protocols are exacerbated by the ease with which attackers can monitor radio signals and passively capture packets as compared to LAN or other physical networks. The intent of this research is to analyze the feasibility of designing a wireless authentication protocol, which is secure against dictionary attacks, for home networks and small wireless networks without using PKI or transport layer security. The research focuses mainly on pre-shared key authentication mechanisms in order to reduce the overhead of directory servers or radius based authentication mechanisms.

CHAPTER 1. INTRODUCTION

This chapter introduces the research by presenting the problem statement and explaining the background that led to this research. This chapter concludes by defining the scope and significance of this project.

1.1. Background

IEEE 802.11 is a set of standards carrying out wireless local area network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands ("IEEE 802.11", n.d). For the purpose of this research, 802.11x represent the standards like 802.11a, 802.11b, 802.11g etc, which are standard wireless protocols used in wireless communication. Some of the 802.11x wireless protocols are increasingly targets of offline dictionary attacks. The purpose of this research is to analyze implementations of Extensible Authentication Protocol (EAP) which is a widely used authentication protocol in 802.1x networks (which include both wired and wireless networks). Extensible Authentication Protocol implementations in 802.11x without using PKI or transport layer security are vulnerable to dictionary attacks. Since wireless communication uses radio signals that can be easily monitored by an eavesdropper, it is important to secure wireless protocols against dictionary attacks.

Extensible Authentication Protocol is a key exchange framework that facilitates authentication between client and wireless routers in a wireless network. However, extensible authentication protocol in itself is prone to offline dictionary attack if not implemented in a secure medium. Since wireless communication uses radio signals that can be easily monitored by an

eavesdropper, it is important to secure wireless protocols against dictionary attacks. Enterprise wireless security protocols implement key exchange mechanisms based on extensible authentication protocol under transport layer security (TLS) or by using public key infrastructure (PKI).

PKI is a viable solution for wireless networks in large organizations and enterprises. Enterprise wireless networks use EAP-PEAP, EAP-TLS (Allen & Wilson, 2002), EAP-TTLS etc (Han, 2006), which are variations implementing extensible authentication protocol (EAP) under transport layer security or using certificates (PKI). However, it is not feasible to implement any of the above mentioned enterprise wireless protocols by a "*Home User"* or "*Small Business".* Hence small scale networks use protocols like WEP, WPA-PSK and WPA-CCMP which are implementations of extensible authentication protocol without using PKI. This makes all home networks open to dictionary attack and paves the way for rainbow-table attacks based on know-plain-text attack, in the future.

Wireless protocols like WEP (Borisov, GoldBerg, & Wagner, 2001) and WPA-PSK (Beck & Tews, 2008) have been subject to a number of different attacks, which try to take advantage of the weakness in underlying encryption/security mechanism. The intent of this research is to analyze and see if dictionary attacks can be avoided on wireless protocols and hence provide a medium for secure wireless computing for small businesses and home users.

1.2. Research Question

Can wireless client authentication in small/home networks be performed without exposing nonce and without using PKI in order to circumvent dictionary attacks?

1.3. Scope

This research is limited to analyzing 802.11x implementations of the EAP key exchange mechanism that facilitates dictionary attacks on wireless security

protocols used in home and small business networks. The scope of this project mainly encompasses analysis of the four way handshake between access point and client station as part of wireless client authentication.

1.4. Significance

The significance of this research can be explained by the fact that the majority of wireless authentication protocols used in home and small business networks are prone to offline dictionary attacks.

Commercially used wireless authentication protocols implementing PKI or transport layer security are the only protocols that currently provide security against dictionary attacks. However, implementations of these protocols are technically intensive, expensive and often have to incur the load of directory services or directory based authentication mechanisms, which are not feasible in home networks or small business networks. This research focuses on designing a robust and secure wireless authentication mechanism that can be made available and affordable to the small scale wireless networks with increased security as compared to current protocol options. The significance of this research can be rightly summarized by the question “Can we design a protocol, without using PKI or complicated infrastructure, that can be implemented in an off the shelf wireless router and can be installed and used by a person without prior technical training or wireless technology background?”

1.5. Assumptions

The assumptions:

- This research assumes that dictionary attack and rainbow table attacks will become more feasible in future, with the increase in computing power and storage facilities. Hence analyzing the gap and securing implementations of key exchange based on extensible authentication

protocol framework against these attacks is necessary to ensure safe and affordable wireless security.

- This research assumes that it is infeasible to brute force a 128 bit encryption key, even with current computing power.
- This research assumes that finding discrete logarithm of a 1024 bit number in reasonable time is a hard problem and cannot be solved using current computing power.
- This research assumes that it is possible to generate a perfectly random number or pseudo-random number using current computing power and available resources.

1.6. Delimitations

This research is performed acknowledging the following delimitations:

- This research analyzes dictionary attack vulnerability in 802.11x implementations of extensible authentication protocol based key exchange mechanism.
- This research does not analyze feasibility of performing dictionary attacks against extensible authentication protocol key exchange mechanism implemented under transport layer security.
- This research analyzes only 802.11x wireless protocols and does not address other 802.1x protocols like point-to-point protocol (PPP) implementing extensible authentication protocol.

1.7. Limitations

This research is limited by the following:

- This research does not address implementation of proposed solutions or protocols.
- This research does not provide implementation benchmark values for performance of proposed solutions or protocols.

1.8. Summary

This chapter highlighted the research contained within this thesis and outlined the importance of analyzing the identified gap. This chapter also highlighted the assumption, limitations and delimitations pertaining to the scope of this research.

CHAPTER 2. LITERATURE REVIEW

This chapter analyzes literature pertaining to 802.11x authentication and key exchange mechanism. The intent of this literature review is to point out the security loop hole in the 802.11x key exchange architecture that facilitates dictionary attack against wireless authentication protocols.

2.1. Dictionary Attack

In cryptanalysis, a *Dictionary Attack* can be defined as a subset of the normal brute force attack. In this method, the attacker tries to decrypt a cipher or defeat an authentication mechanism by trying all possible words, from an offline dictionary file, as the *encryption/decryption key.* The effectiveness of a mounted dictionary attack is determined by the strength of the dictionary used. If the password or key is chosen such that the chosen key is not present in any standard dictionary, dictionary attacks can be avoided (“Dictionary Attack,” n.d.). However, an exhaustive and thorough dictionary, also termed as *Rainbow Table,* of all possible alpha-numeric characters is not a farfetched reality.

Dictionary attacks are a time-memory tradeoff (“Dictionary Attack,” n.d.). With the advent of super fast processors and high capacity network storage units, it is not long before dictionary attacks/rainbow table attacks become an achievable reality.

2.2. Analyzing EAP 4-Way Handshake

Wireless connections between a client station (*STA*) and wireless access point (*AP*) include the following steps: ("IEEE 802.11i-2004," 2004):

1. Probe Request
2. Probe Response
3. Authentication Request
4. Authentication Response
5. Association Request
6. Association Response
7. 4-way Handshake

Our main focus is on Step 7, as this is the step that facilitates dictionary attacks on a wireless security protocol.

The 4-way handshake is a key exchange mechanism that facilitates mutual authentication between a client station and wireless access point in a wireless network. As the name implies, the 4-way handshake involves four packets exchanged between the client and wireless access point. These packets are known as the EAP over Lan Key (*EAPOL-KEY)* packets and are crucial towards mounting a successful dictionary attack.

Though the 4-way handshake consists of four packets, an attacker requires only the first two packets to start an effective dictionary attack on the protocol. After obtaining the first two EAPOL-KEY packets, the attacker has knowledge of the nonce values (ANonce, SNonce – discussed in following chapters), which are pseudo-random numbers generated by the wireless access point and client station respectively, and dictionary words, which are required to launch a successful dictionary attack against the AP (Lehembre, 2005, p. 14). This vulnerability is discussed in detail in the coming sections.

2.3. Analysis of WPA-PSK (TKIP) 4-Way Handshake

WPA-TKIP is one of the most commonly used wireless security protocols in home networks and small businesses.

WPA-TKIP implements the extensible authentication protocol based key exchange mechanism described in section 2.2 of this chapter. However, unlike other implementations of extensible authentication protocol, WPA assumes that both client station and access point have prior knowledge of a shared key known as the *Pairwise Master Key (PMK)*. PMK is generated from the WPA passphrase or pre-shared key (*PSK*) provided by the end user, using the following function (Rantwijk, n.d):

PMK = PBKDF2 (passphrase, ssid, 4096, 256)

This function takes the passphrase provided by the end user and name of the wireless network (SSID) and perform HMAC-SHA1 on them 4096 times to obtain a 256 bit Pairwise Master Key (PMK). The difference between PMK and PSK is that PSK is a random word chosen by the user and PMK is a 256 bit secure key generated from the PSK using name of the wireless network (SSID) as a seed. The purpose of this step is to ensure that no attacker can pre-compute all possible PMK's using dictionary words as PSK's. Since the name of wireless networks (SSID) changes according to user and location, using the SSID as seed ensures that one large offline dictionary cannot be used against all networks.

Once PMK is calculated as discussed in the above step, the client station (STA) and wireless access point (AP) start the four-way handshake. The four-way handshake is initiated by the AP and proceeds as shown in figure 2.1. A detailed representation of 802.11i key exchange mechanism is shown in Figure 2.1, depicting the key exchange mechanism between Client station (also known as Supplicant) and Wireless access point (also known as Authenticator).

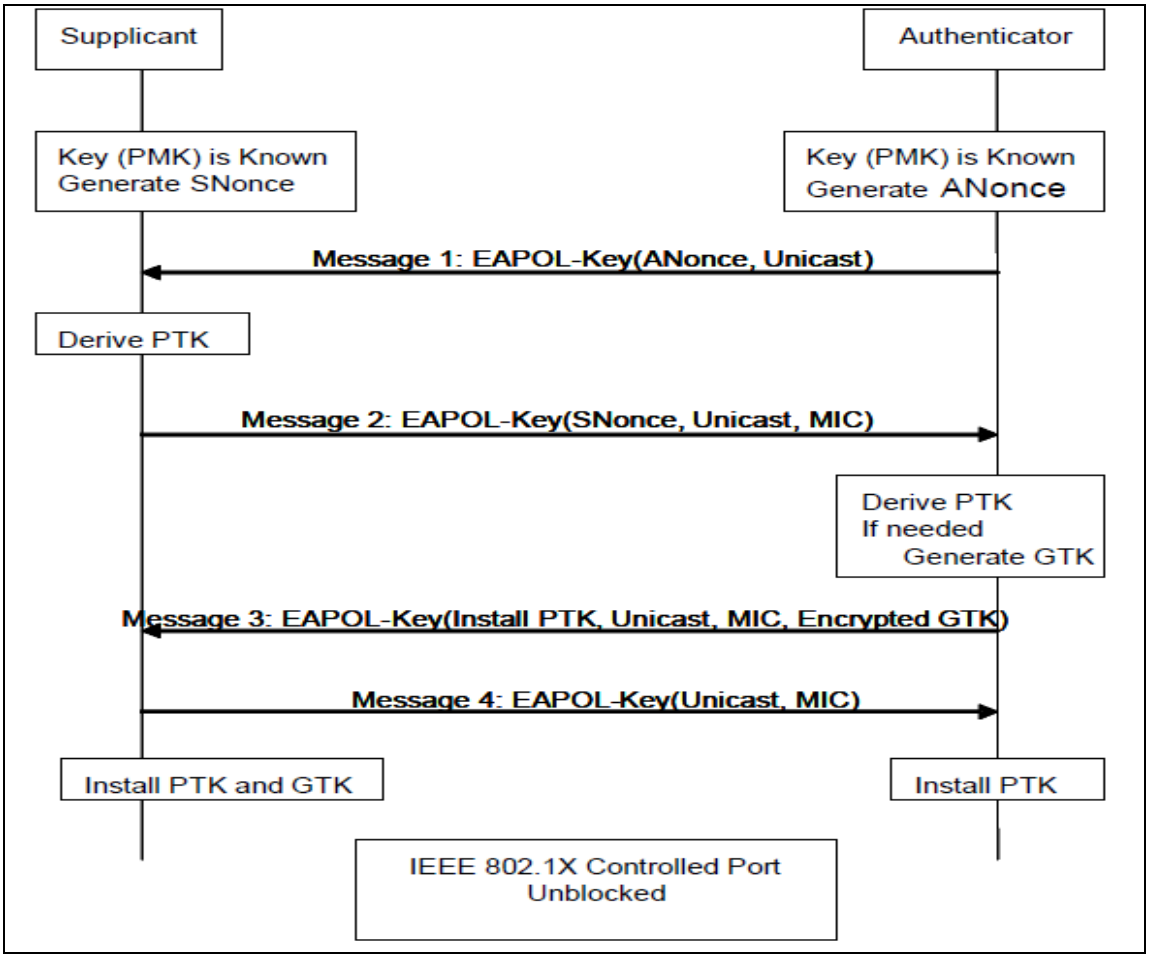

Figure 2.1 EAP 4-way handshake, Source "IEEE 802.11I-2004: ", 2004, p. 16.

1) Step 1: Wireless access point (AP) generates a random nonce, ANonce, and sends it in clear text to the client station (STA).
2) Step 2: The client station (STA) receives ANonce and generates nonce, SNonce. Client station (STA) then generates the *Pairwise Transient Key* (PTK) by concatenating Pairwise Master Key (PMK) with ANonce, SNonce, AP‟s MAC address & STA‟s MAC address and putting the concatenated output through a cryptographic hash function. Pairwise Transient Key (PTK) acts as a session key that facilitates secure communication between client and access point for that particular session

of communication. Client station (STA) then generates a packet containing SNonce in clear text and computes message integrity check value (MIC) over the packet using a key TxK derived from Pairwise transient key(PTK) as shown in figure 2.2. Client station (STA) then transmits the SNonce and computed MIC to the wireless access point (AP) as part of this step. The generation of PTK is shown in Figure 2.2 (Lehembre, 2005).

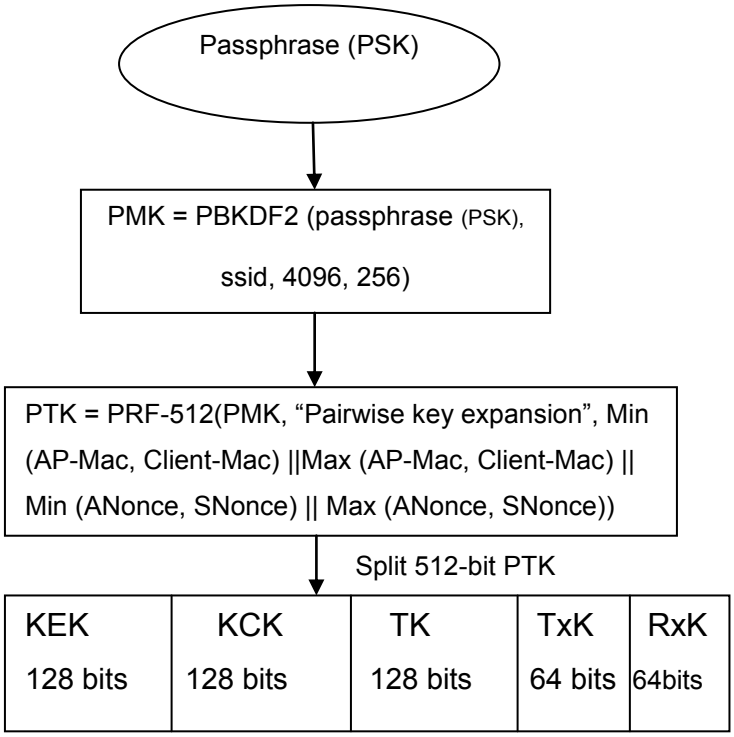

Figure 2.2 WPA PTK generation.

3) Step 3: Wireless access point (AP) generates GTK and sends GTK+MIC to client station (STA). The generation of GTK and MIC over GTK is not discussed in detail as this research focuses mainly on step 1 & 2 of the 4-way handshake which are essential for mounting an offline dictionary attack.
4) Step 4: Client station (STA) sends back acknowledgement ACK to wireless access point (AP).

2.4. Cryptanalysis of WPA

Pairwise Master Key (PMK) generation of WPA is the most time consuming step as it involves 4096 iterations of HMAC-SHA1. Because PMK is dependent on SSID, it is infeasible to generate a dictionary that has the PMK for all possible SSIDs.

Commonly used tools such as cowpatty (*coWPAtty,* n.d) and aircrack-ng (*Aircrack-ng ,* n.d) generate pre-computed PMK dictionaries from ordinary dictionaries to mount attacks on WPA networks. These dictionaries consist of pre-computed PMK‟s that are computed over normal dictionary words and SSID provided by the user. Once the pre-computing is performed, the attacker captures the 4-Way handshake between any client station (STA) and targets wireless access point (AP) by passive monitoring or actively de-authenticating a client.

After capturing the 4-way handshake, the attacker has a(n):

1) Pre-computed PMK dictionary,
2) ANonce,
3) SNonce ,
4) MAC Address of wireless access point (AP) and client station (STA) obtained from the packets, and
5) Message integrity check value (MIC) calculated over SNonce packet (from STEP 2 of EAPOL-KEY Exchange)

2.4.1. The Summary of Attack

The step wise summary of dictionary attack on WPA-PSK performed by an attacker is as follows:

1) Choose pre-computed PMK from the dictionary
    a. Compute PTK using PMK, Nonce values (ANonce, SNonce) and MAC Addresses as explained in section 2.3
    b. Generate key TxK from PTK as shown in figure 2.2

c. Calculate message integrity check value (MIC) using TxK over SNonce value passively monitored by attacker and compare with MIC value passively observed by the attacker in STEP 2 of EAPOL-KEY exchange (4-way Handshake)
d. If computed MIC value matches with captured MIC go to next step, else go to STEP 1

2) If there is a MIC match, return associated PMK, else return failure.

Knowledge of wireless network name (SSID) and pseudo-random numbers (Nonce values – ANonce, SNonce) facilitates an attacker to perform a *known-plain-text* attack against WPA. SSID is an inevitable part of wireless discovery process and hence cannot be hidden for long. However, nonce, if hidden from plain view, can certainly defeat a dictionary attack based on *known-plain text* attack.

2.5. Analysis of MSCHAPv2 4-Way Handshake

This section analyzes dictionary attack on version 2 of Microsoft‟s Challenge-Handshake Authentication Protocol (MSCHAPv2). MSCHAPv2 analysis is used as a case study to analyze the similarities in dictionary attack on different wireless authentication protocols.

MSCHAPv2 implements the standard key exchange mechanism based on extensible authentication protocol, which has been discussed in section 2.2 of this thesis (*IEEE 802.11i*, 2004).

The 4-way handshake in MSCHAPv2 is initiated by the AP and proceeds as follows:

1) Step 1*:* Wireless Access point (AP) generates a 16 byte random nonce (ANonce) and sends it in clear text to the client station (STA)
2) Step 2: The client station (STA) receives ANonce send by the wireless access point and generates its own nonce, SNonce. It then generates an

NT-Response value using SNonce, ANonce, username and password (Zorn, *RFC2759*, n.d.). Client station (STA) then transmits the NT-Response, SNonce and username in clear text to the AP.

3) Step 3: Wireless access point (AP) verifies the *NT-Response* it receives from the client station STA. If verification succeeds, the wireless access point (AP) sends a *GenerateAuthenticatorResponse* packet to the client station (STA). Verification of NT-Response and generation of GenerateAuthenticatorResponse are beyond the scope of this thesis as this research focuses in Steps 1 & 2 of 4-way handshake.
4) Step 4: Client station (STA) verifies the *GenerateAuthenticatorResponse* it receives from the wireless access point (AP). If the verification succeeds, the client station (STA) sends back an ACK to the wireless access point (AP).

2.6. Cryptanalysis of MSCHAPv2

As explained by Joshua Wright in *asleap* readme (Wright, n.d.):

The MS-CHAPv2 challenge/response suffers from several notable flaws:

- No salt is used in conjunction with the NT hash : Permits pre-computed dictionary attacks
- Weak DES key selection for challenge/response : Permits recovery of 2 bytes of the NT hash
- Username sent in clear-text

Hence MSCHAPv2 can be attacked in two ways:

- An ordinary dictionary attack can be performed against MSCHAPv2
- A pre-computed dictionary attack can be performed, taking advantage of the weak DES encryption mechanism used, wherein the *NT hash* of the dictionary passwords can be pre-computed

In both the cases mentioned above, knowledge of nonce values (challenge) and username is crucial for performing a successful dictionary attack. The knowledge of nonce (challenge) and username transmitted in clear text along with the knowledge of cipher-text (NT-Response) makes it possible to perform a *known-plain-text* attack against MSCHAPv2.

2.7. Conclusion

As observed in the cases of WPA and MSCHAPv2, the common factors noticed in relation to dictionary attacks are:

1) ANonce - Nonce generated by wireless access point (AP)
2) SNonce - Nonce generated by client station (STA)
3) Dictionary attack is effective after EAPOL-KEY exchange step 2

In all of the cases discussed above, nonce is crucial for performing a *Known-plain-text* attack using a dictionary. Because nonce is transmitted in clear text and the attacker can monitor the EAPOL-KEY exchange to get the corresponding *Cipher Text*, an attacker can successfully perform dictionary attacks on the network.

Nonce/Challenge is a crucial factor that determines whether a dictionary attack can be successfully performed against an 802.11x protocol. Though not an immediate danger, dictionary attacks pave the way for rainbow table attacks, which are not far from reality. Based on the analysis in this chapter, we observe that dictionary attacks can be averted by ensuring that the attacker cannot mount a *known-plain-text* attack on any protocol. This can be attained by ensuring that the nonce/challenge is not transmitted in clear text. If the plain-text is *unknown* (nonce/challenge in these cases), the attacker can only attempt *cipher-text-only* attack, which tries to take advantage of vulnerabilities of the underlying encryption mechanism.

2.7.1. Known-Plain-Text Attack Summary of WPA

The summary of known-plain-text attack on WPA-PSK is as follows:

Plain-Text: SNonce

Cipher-Text: MIC calculated over SNonce using TxK

2.7.2. Known-Plain-Text Attack Summary of MSCHAPv2

The summary of know-plain-text attack on MSCHAPv2 is as follows:

Plain-Text: Challenge (ANonce, SNonce) + Username

Cipher-Text: NT-Response.

2.8. Summary

This chapter analyzed existing literature on the 802.11x authentication mechanism. Two of the widely used wireless security protocols were considered for case study in order to determine the nature of dictionary attacks on wireless protocols.

Though attempts have been made to secure the key exchange mechanism using PKI, it is not a viable mechanism that can be implemented on a small scale (home networks or small scale businesses). None of the literature or existing protocols have addressed ways to secure nonce exchange. Also this review confirms that attempts to secure key exchange using PKI still retain the inherent flaw in the key-exchange framework within the secure tunnel established using PKI.

Additionally, this chapter has emphasized the need to secure nonce exchange as part of EAP key exchange mechanism, which is discussed in detail in the following chapters.

CHAPTER 3. FRAMEWORK AND METHODOLOGY

The purpose of this research is to analyze the structure and functioning of extensible authentication protocol (EAP) based wireless key exchange mechanisms and identify the root-cause that facilitates dictionary attacks against protocols designed with EAP as its core.

3.1. Theoretical Framework

Extensible Authentication Protocol (EAP) is a universal authentication framework used in most of the 802.11x wireless authentication protocols. EAP, being a framework, defines only message formats and hence encapsulation and security of messages is left to implementation of the protocol adopting EAP. There have been attempts to design wireless security protocols, without using PKI, for small home networks. Examples of such protocols are WEP, WPA-TKIP and WPA-CCMP. However, all these protocols implementing EAP without encapsulation or transport layer security have been found to be vulnerable to offline dictionary attacks.

As explained in chapter 2, nonce exchange takes place between the client and access point during the four-way handshake of extensible authentication protocol (EAP) based key exchange. These nonces help in increasing the randomization of the encryption algorithms and facilitate the client to access-point authentication and vice versa. The nonce values also ensure that eaves droppers cannot replay a captured handshake again as the nonce values are random. Though nonces play a vital role in encryption randomization and

avoiding replay attacks, the nonce exchange has resulted in a potential weakness in the framework that facilitates offline dictionary attacks.

This research aims at analyzing the feasibility of securing implementations of extensible authentication protocol (EAP) wireless key exchange against dictionary attacks. An attacker can ideally capture the first two packets of the extensible authentication protocol (EAP) based wireless key exchange and successfully initiate a dictionary attack. This is facilitated by the fact that nonce exchange happens in clear text. Hence, once the attacker has the clear text nonce and associated cipher text generated by the client, using the nonce, as part of authentication process, the attacker can perform a known-plain-text cryptanalysis on the cipher text until it yields the plain text.

The approach to this problem should be one that ensures that nonce exchange of extensible authentication protocol (EAP) based wireless key exchange mechanism does not facilitate known-plain-text attack on authentication protocols. This can be achieved by avoiding transmission of nonce values in clear text. This would not only increase the randomness as described earlier, but also increase the level of security.

3.2. Research Framework

This thesis explores and presents a study of security and encryption principles that can be applied to the implementations of extensible authentication protocol (EAP) based wireless key exchange in order to make it secure against dictionary attacks. The research aims at analyzing the feasibility of securing implementations of extensible authentication protocol (EAP) based wireless key exchange using the following methods:

- Diffie-Hellman key exchange to secure nonce
- Encrypting nonce using randomized master key
- Using Hash function to generate randomized keys that can encrypt nonce

The effect of these methods will be analyzed across the following areas:

- Security against Man-In-Middle attacks
- Security of master key or password while securing nonce
- Quality of randomization in nonce encryption

The research will focus on analyzing whether it is feasible to secure implementations of extensible authentication protocol (EAP) based authentication protocols against dictionary attacks without using PKI or transport layer security.

H_0: It is not feasible to secure EAP based authentication protocols against dictionary attacks without using PKI

H_α: It is feasible to secure EAP based authentication protocols against dictionary attacks without using PKI

3.3. Methodology

This section explains the methodology adopted in analyzing the security of the extensible authentication protocol (EAP) based key exchange by applying the methods mentioned in section 3.2 of this thesis.

3.3.1. Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange mechanism can be used to ensure that nonce values are encrypted before transmission, between client and access-point, without prior knowledge of encryption key to each of the participants ("Diffie-Hellman Key Exchange," n.d). This ensures that an eaves dropper does not have access to plain-text nonce values and hence cannot perform a known-plain-text dictionary attack against the authentication protocol.

Though the Diffie-Hellman key exchange ensures authentication, it does not have any provision to ensure integrity of transmitted data. This facilitates a Man-In-The-Middle attack on the key exchange mechanism. Hence, this

research focuses at analyzing the feasibility of applying the Diffie-Hellman key exchange mechanism to negotiate a key that can be used to encrypt the nonce values and at the same time ensure integrity of data transfer.

3.3.2. Encryption Using Randomized Master Key

The research will focus on analyzing the feasibility of randomizing the master key and encrypting the nonce values using the new randomized key. Key randomization process should be adopted such that it facilitates both client and access point to derive the same session key used for encrypting the nonce.

This part of research also focuses on security of key randomization process with respect to security of master key. The goal of the research is to ensure security without exposing any part of master key during the process of authentication.

3.3.3. Hash Functions to Generate Random Keys

This research will analyze the feasibility of applying hash functions to generate random session keys that can be used to encrypt nonce exchange. The initialization vector of the hash function should facilitate the client and AP to generate a common hash that can be used as the key for encryption and decryption.

Security of master key is a prime area of research pertaining to this method of securing extensible authentication protocol (EAP) based key exchange mechanism.

3.4. Summary

This chapter has provided an overview of the framework and methodology used in this research. It describes the methods under analysis and its impact on the security of extensible authentication protocol (EAP) based key exchange

mechanism. The next chapter will highlight the outcome of the analysis mentioned in this chapter and analyze the feasibility of securing extensible authentication protocol (EAP) based key exchange mechanism from dictionary attacks.

CHAPTER 4. SOLUTIONS AND OBSERVATIONS

As explained in chapter 2 of this thesis, exposing the nonce values during wireless authentication mechanism paves way for dictionary attacks. Dictionary attacks against the wireless authentication mechanism for home or small networks can be avoided by ensuring that the adopted wireless authentication mechanism can overcome known plain text attacks.

4.1. Using the Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange mechanism was one of the proposed mechanisms to ensure that nonce values are not exchanged in clear text during wireless key authentication mechanism. However, Diffie-Hellman does not provide authentication as part of the key exchange mechanism and hence is vulnerable to man-in-the-middle (Schneier, n.d) attacks.

Since wireless authentication systems under consideration in home and small business networks use pre-shared key (PSK) as part of authentication mechanism, this feature can be extended to the Diffie-Hellman key exchange mechanism to yield an authentication mechanism based on password-authenticated key agreement (PAKE). The researcher discusses an implementation of wireless authentication mechanism for home and small business networks using augmented password-authenticated key agreement mechanism called *Secure Remote Password Protocol* in section 4.2.

4.2. Using Secure Remote Password Protocol

Secure Remote Password protocol (SRP) is an augmented password-authenticated key agreement (Augmented PAKE) mechanism widely used for authentication. The key features of this protocol that makes it appealing to this research are:

1. It is resistant to dictionary attacks, and
2. It does not require trusted third party and hence does not rely on PKI

Implementations of Secure Remote Password Protocol (SRP) as part of Extensible Authentication Protocol (EAP) for wireless authentication has already been proposed as part of EAP-SRP-256 protocol (Manganaro, Koblensky, & Loreti, 2009) and EAP SRP-SHA1(Carlson, Aboba, & Haverinen, 2002). However, these SRP implementations of EAP based authentication mechanisms rely on password based authentication at a per user level. Hence these implementations cannot be easily adapted to home or small business networks that do not have directory services or radius (Congdon, Aboba, Zorn, & Roese, 2003) based user authentication mechanism. The focus of this research is to design a simple authentication mechanism that can be implemented at home and small business wireless networks.

The new EAP method proposed in this thesis is named *EAP-SRP-PSK*. This Extensible Authentication Protocol based authentication mechanism aims at providing a secure authentication mechanism by using pre-shared key architecture used in earlier solutions for home or small business networks.

4.2.1. EAP-SRP-PSK Specification

EAP-SRP-PSK is designed based on SRP-6 specification (“SRP Design Specification,” n.d). The specifications of this protocol design are an extension of SRP-6 specification. The following abbreviations adopted from SRP-6

specification are components used as part of EAP-SRP-PSK‟s authentication mechanism:

N	:	A large 1024 bits Safe Prime number
g	:	A generator modulo N
a, b	:	Two 128 bit secret random numbers used to generate A, B
A, B	:	Public exchange keys
s	:	512 bit session pseudo random number
x	:	512 bit session private key
v	:	Password verifier
k	:	Multiplier parameter
SK	:	Session key
SKV	:	Session key verifier
HM2	:	HMAC-SHA-512 HMAC function
H2	:	SHA-2 (512) one way hash function
u	:	Hash of (A,B)
IC1, IC2	:	Integrity Checks
PSK	:	Pre-shared key
PMK	:	Pairwise Master Key
SSID	:	Service Set Identifier, Name broadcasted by the AP

4.2.1.1. PMK Generation

Pre-Shared Key (PSK) is a shared password manually configured and initialized in the access point and shared/communicated amongst all users who wish to use the network. All users connecting to the access point uses the same PSK for authentication purpose. Knowledge of the pre-shared key (PSK) is hence an essential part of the authentication scheme. Configuration and resetting of the pre-shared key (PSK) is performed by administrators by reconfiguring the

access point either over wired network or wireless network, depending on the network configuration.

To avoid any possible pre-computation attacks on the protocol, EAP-SRP-PSK generates PMK from the user defined pre-shared key (PSK) by using a variation of PBKFD2 method used in WPA-PSK:

PMK = PMKGen(PSK, SSID, 4096,512)

This function takes PSK and SSID as the input and computes its hash value by performing HMAC-SHA-512 4096 times to output a 512 bit PMK.

This step is a onetime process and is performed by the access point when it boots or when the PSK value is changed by the user. Client machines perform this operation once before establishing connection with the access point.

4.2.1.2. Authentication Mechanism / Handshake

- Client sends authentication request to access point and specifies protocol as EAP-SRP-PSK
- Access Point generates N and g values:
    - N is a 1024 bit large safe prime number
    - g = (small generator prime number) modulo N
    - Access Point sends (N,g) pair along with keyword “Generators” to client as reply to client authentication request
- Client performs the following operations:
    - Chooses a random private key *a*.
    - Client calculates $A=g^a$ mod N and send it across to server along with keyword “*client public key”*.
- Access Point chooses a 512 bit random salt „*s’* and performs the following calculations:
    - Access point receives A from client. If A = 0, abort connection else proceed with next step.

- x = HMAC-SHA-512(s, PMK), outputs a 512 bit HMAC output using s and PMK as input.
- $v = g^x \bmod N$.
- k = SHA-512(N, g), generate 512 bit SHA-2 hash from N and g values.
- Choose a random private key b.
- Calculate $B = (kv+g^b) \bmod N$.
- Access point sends (s,B) pair to the client along with key word "*ap public key"*.

The above four steps conclude the 4-way Handshake and enables the client and access point to authenticate and verify each other and negotiate a session key using the exchanged parameters. If B=0, Client will reset connection and re-initiate connection. Section 4.2.1.3 discusses generation of the session key from the exchanged parameters and verification procedure in detail.

4.2.1.3. Session Key Generation

After successful completion of the four-way handshake, client performs the following calculations to obtain the session key:

- x = HMAC-SHA-512(s, PMK), outputs a 512 bit HMAC output using s received from the server and PMK as input.
- u = SHA-256(A, B), 256 bit hash value of A, B.
- SK (Session Key) = $(B - kg^x)^{a+ux} \bmod N$
- SKV (Session key verifier) = SHA-512(SK), 512 bit session key verifier.

Access point computes the session key as follows:

- u = SHA-256(A, B), 256 bit hash value of A, B.
- SK (session key) = $(Av^u)^b$.
- SKV (Session key verifier) = SHA-512(SK), 512 bit session key verifier.

Once the client and access point have generated the session key, the session key verifier (SKV) can be used to verify the integrity of the client and access point as follows:

- Client computes integrity check value IC1:
    - IC1 = SHA-512(s, B, SKV).
    - Client sends IC1 to Access point along with message “Client *Integrity Check”.*
- Access point verifies value of IC1, if verification succeeds, calculate IC2:
    - IC2 =SHA-512(A, IC1, SKV).
    - Access Point sends IC2 to client for verification along with message “*AP Integrity Check”.*

After successful verification, the client and access point are now authenticated and verified and possess a session key that allows secure traffic between access points and client stations. Access point stores the client systems MAC address and session key pair for further communication between client and access point.

A complete representation of the 4-way handshake and authentication process is as depicted in figure 4.1.

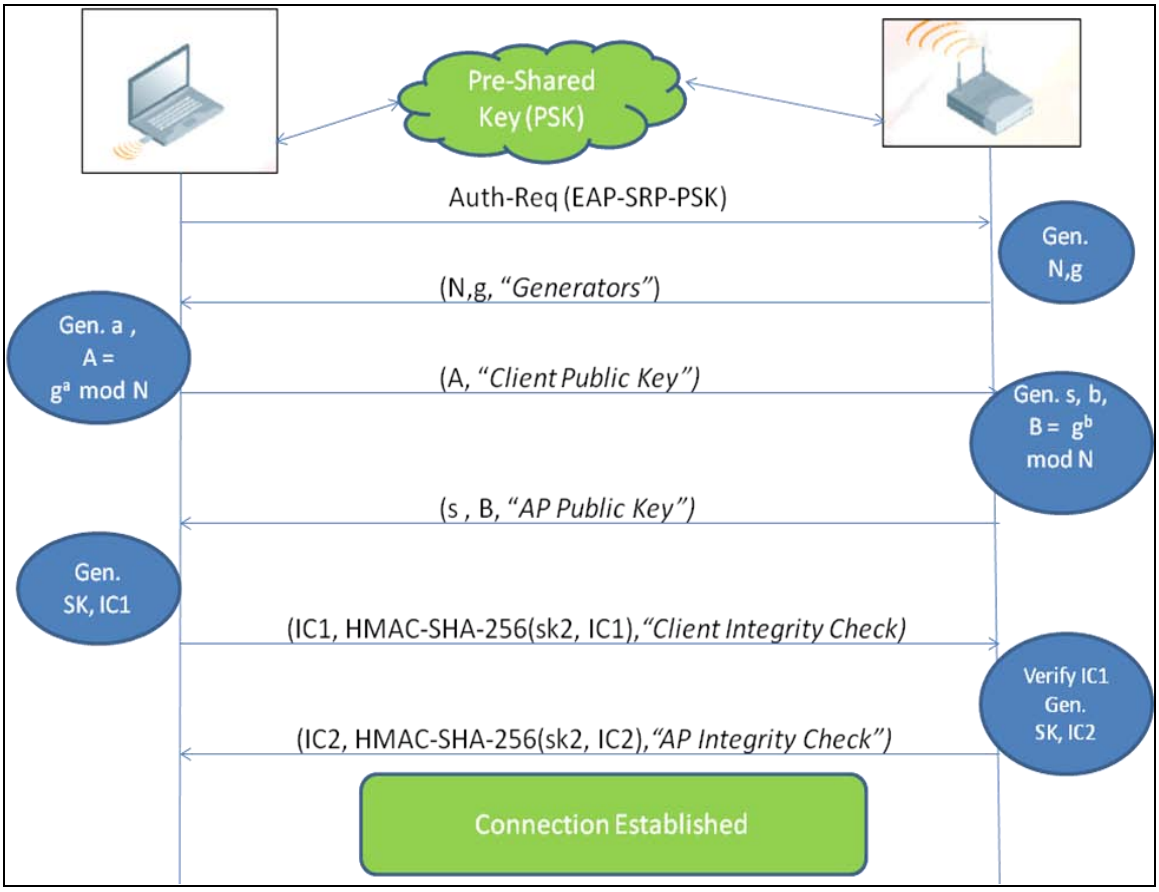

Figure 4.1: EAP-SRP-PSK Authentication.

4.2.1.4. Key Generation from Session Key

Section 4.2.1.3 discussed the generation of session key by the client and access point. The session key (SK) generated by the client and access point is 1024 bits long.

4.2.1.4.1. Unicast Encryption Key (EK)

EAP-SRP-PSK uses AES-256 to encrypt all unicast data packets between the client system and access point. For the purpose of using AES-256, the system generates a 256 bit Encryption key hereby abbreviated as EK as follows:

- Split 1024-bit session key SK in to 4 equal parts sk_1, sk_2, sk_3 and sk_4. Each of sk_i (for i in 1,2,3,4) is 256-bit long.

- Generate EK:
  - EK = sk_1 + sk_2 + sk_3 + sk_4, where + represents XOR operation.

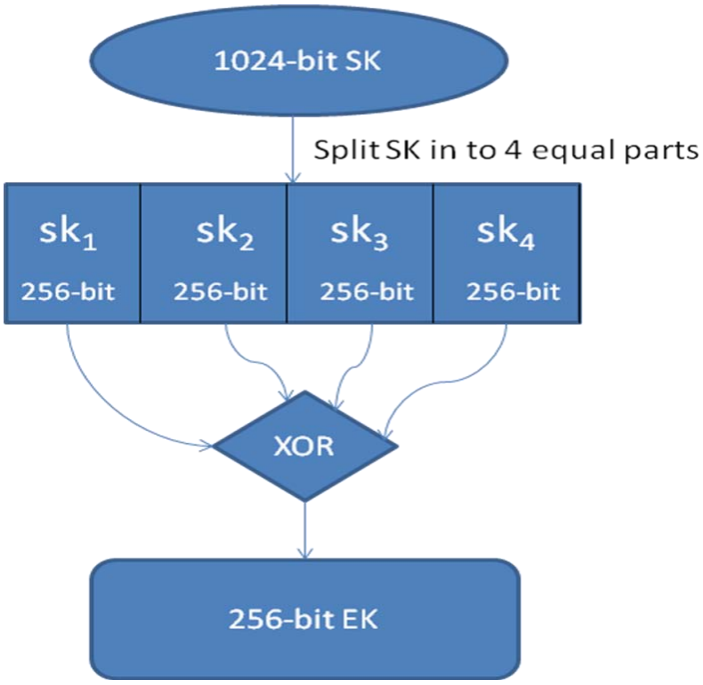

Figure 4.2 Encryption key (EK) generation.

4.2.1.4.2. Integrity Assertion Key (IAK)

Integrity Assertion Key abbreviated as IAK is used by the client and access point to prove their identity and integrity during the authentication process. IAK is used to compute HMAC-SHA-256 message integrity value over IC_1 and IC_2 integrity check values generated by the client and access point respectively. This helps to avoid any *Forced Reset Attac*k on the protocol.

Forced Reset Attack (FRA) can be performed by an attacker by sending a fake IC_x value to the client or access point. On receiving a fake IC_x value, the station resets connection since the fake IC_x does not match the authentic IC_x generated by the station. This attack can also be used to perform *Denial of Service* (DOS) attacks against the access point by flooding the access point with fake IC_x values for different MAC addresses trying to establish connection with

the access point. Incorporating the message integrity check using Integrity Assertion Key (IAK) helps the access point and client stations to detect and weed out Forces Reset Attack.

Integrity Assertion Key (IAK) is a 256 bit key derived from Session Key (SK). IAK corresponds to bits at positions 257-512 of Session Key (SK) represented as sk_2 in Figure 4.2. IAK is used as a key in HMAC-SHA-256 along with IC_x to yield a message integrity check value that is transmitted along with IC_x packets.

4.2.1.4.3. Message Integrity Key (MIK)

Message Integrity Key abbreviated as MIK is used to compute message integrity check value over unicast packets transmitted between the client and access point. MIK is used to compute 256-bit HMAC-SHA-256 integrity values over unicast packets. MIK is a 256-bit key obtained from bits 513-768 of Session Key (SK), represented as sk_3 in Figure 4.2.

Message integrity check value is computed by supplying MIK as key to HMAC-SHA-256 function along with 256-bit block of data encrypted using AES-256 with encryption key EK (section 4.2.1.4.1).

4.2.1.4.4. Key Encryption Key (KEK)

Key encryption key (KEK) servers the same purpose KEK"s played in WPA-PSK. KEK is used to facilitate encrypted exchange of Group Temporal Keys between access points and client machines. KEK is a 256-bit key generated from bits 769-1024 of Session Key (SK), represented as sk_4 in Figure 4.2.

KEK is used as encryption key for encrypting and distributing Group Temporal Key"s to all client machines. Integrity Assertion Key (IAK) is used to compute message integrity check value over packets encrypted using KEK, before transmitting to the client.

4.2.1.5. Connection Reset

Connection reset is different from *Forced Reset Attack* described in section 4.2.1.4.2 of this thesis. Connection reset is a safety precaution adopted by the Access Point to ensure that faulty connection requests are dropped to ensure integrity of the protocol.

A connection reset is issued by the client or access point to notify its peer that all values and keys associated with the session are being reset as the client/ access point encountered an unexpected scenario during the handshake process. The scenarios during handshake that could result in a connection reset are:

- Access Point issues a connection reset if public Key A issued by the client is 0.
- Client issues a connection reset if public key B issued by the access point is 0.
- Access Point issues a connection reset if Integrity Check Value IC_1 issued by the client does not match the IC_1 value generated by the access point.
- Client issues a connection reset if the integrity check value IC_2 issued by the access point does not match the IC_2 value generated by the client.

4.2.2. EAP-SRP-PSK Protocol Security

This section discusses the security provided by EAP-SRP-PSK protocol key exchange mechanism.

4.2.2.1. Theorem 1: Why does EAP-SRP-PSK work

EAP-SRP-PSK uses SRP-6 to generate a unique session key that can be used for secure communication. The key exchange mechanism neither exposes private key of the client nor of the access point. However, the protocol enables both parties with knowledge of *Shared Password* to derive a secret session key successfully.

Theorem Statement: "EAP-SRP-PSK protocol allows client station and access point to derive same session key"

Proof:

The following proof explains how client station and access point independently derives the session key using EAP-SRP-PSK protocol.

Client Calculation:

After the first four steps of authentication mechanism, the client possesses the N, g, a, A, B, s and PMK which were exchanged as part of the key exchange mechanism. At this point, the client has knowledge of only the Public Key of the access point (B) and not the private key. Using these known values, the client performs the following calculations which explain how client and access point procures the same session key:

- Client calculates x = HMAC-SHA-512(s,PMK)
- Client calculates u = SHA-256(A, B)
- Calculate SK = $(B - kg^x)^{a+ux} \bmod N$

  $= (kv + g^b - kg^x)^{a+ux} \bmod N$, because $B = (kv + g^b) \bmod N$

  $= (kg^x + g^b - kg^x)^{a+ux} \bmod N$, because $v = g^x \bmod N$

  $= g^{b(a+ux)} \bmod N$

Access Point Calculation:

After the first four steps of the authentication mechanism, the access point possesses N, g, s, b, A, B, s and PMK values. The access point has knowledge of only the Public Key (A) of the client and not the private key. Using these values the access point performs the following calculations to derive the same key as obtained by the client:

- Access Point calculates x = HMAC-SHA-512(s,PMK)
- Access Point calculates $v = g^x \bmod N$
- Access Points calculates u = SHA-256(A, B)
- Calculate SK = $(Av^u)^b \bmod N$

  $= (g^a.v^u)^b \bmod N$, because $A = g^a \bmod N$

  $= (g^a. g^{xu})^b \bmod N$, because $v = g^x \bmod N$

$= (g^{a}. g^{xu})^{b} \bmod N$, because $v = g^{x} \bmod N$

$= g^{b(a+ux)} \bmod N$

As shown in the calculations above, both client and access point derive the same session key ($g^{b(a+ux)} \bmod N$) with knowledge of only the shared information along with the peers public key, hence proved.

Since the private keys a and b are essential for calculation of session key (as shown above), any thirdparty without explicit knowledge of atleast one private key cannot obtain unauthorized access. At no point in time, neither client nor access point is required to share their private keys as part of the protocol. This makes the protocol resistant to passive monitoring for obtaining the keys

4.2.2.2. Theorem 2: Security against Man-In-The-Middle Attacks

EAP-SRP-PSK provides security against Man-In-Middle attacks. Though an adversary may be able to masquerade as a legitimate user during the initial phases of key exchange, the message integrity checks values IC_x generated from session key SK would not match the actual value generated by the client/access point.

Theorem Statement: "EAP-SRP-PSK provides security against Man-In-Middle attacks"

Proof:

Adversary masquerading as Client:

Suppose an adversary chooses a random private key a" and tries to masquerade as a legitimate client. The adversary can initiate an authentication mechanism and obtain a values N, g, B, s from the access point as shown in figure 4.1.

Using the g value obtained from the access point, the adversary can even compute A" as:

$A^{"} = g^{a"} \bmod N$.

However, to compute the session key, the adversary requires B, k, g, a" and u as described in section 4.2.2.1. The adversary can only calculate session key as follows:

- Adversary Calculates x" = HMAC-SHA-512(s, PMK_{wrong}), where PMK_{wrong} is any PMK value generated by the adversary using words from a dictionary
- Adversary Calculates $SK" = (B - kg^{x})^{a"+ux"} \bmod N$
  $= (kv + g^{b} - kg^{x})^{a"+ux} \bmod N$, because $B = (kv + g^{b}) \bmod N$
  $= g^{b(a"+ux)} \bmod N$
- Adversary Calculates IC1" = SHA-512(s, B, SHA-512(SK"))
- Access Point Calculates IC1 for verification, IC1 = SHA-512(s, B, SHA-512(SK))
  - SK" ≠ SK as a" ≠ a & x ≠ x", therefore IC1 ≠ IC1"
- Since IC1 verification fails, Access point rejects transaction and resets connection.

Thus the adversary fails to become a Man-In-Middle or relay agent, hence theorem proved. Thus the adversary cannot compute x without knowledge of PMK, he/she cannot derive the same session key SK as is derived by the access point, This results in an IC_x mismatch followed by a connection reset by the access point.

The same shortcoming is applicable when the adversary tries to masquerade as the access point and tries to negotiate a session key with the client station. The client resets the connection and terminates the transaction when it detects IC_x mismatch between itself and the adversary.

Further, the adversary cannot compute message integrity value over IC_x packets as he/she does not possess the Integrity Assertion Key (IAK) required to generate the integrity check value over IC_x. Since the adversary cannot verify its authenticity, both the client as well as access point reset connection and terminate transaction with the adversary.

4.2.2.3. Theorem 3: Session Key Randomness

The randomness of the session key is ensured by the 512 bit random salt „s" used to generate HMAC value x over PMK.

Theorem Statement: "If salt „s" is truly random, EAP-SRP-PSK produces random session key for every session"

Proof:

Assume „s" to be a pseudo random number which is truly random, i.e.

$Pr[s] = 1/2^{512}$, as „s" is a 512 bit salt.

Session key SK is calculated as $SK = g^{b(a+ux)} \bmod N$

„x" is random because „s" is random, based on our assumption:

x = HMAC-SHA-512(s, PMK)

Hence, Session key SK is random as long as salt „s" is truly random. By ensuring that „s" is truly random and not repeated, every session will have a unique fingerprint and session key that helps avoid pre-computation attacks on the protocol. This ensures that session key generated by EAP-SRP-PSK is random for every session, hence proved.

4.2.2.4. Theorem 4: Security against Dictionary Attack

In a dictionary attack, an adversary uses words from a standard dictionary as pre-shared key and executes the algorithm used by EAP-SRP-PSK protocol to see if the session key derived by the adversary is same as the session key obtained by passive monitoring of wireless traffic, as explained below.

An adversary can obtain values N, g, A, B, s, IC1 and IC2 as shown in figure 4.1, by passively monitoring wireless traffic between the access point and client. Ideally and adversary can perform an offline dictionary attack to obtain PMK as follows:

- Pre-compute all possible values of x : HMAC-SHA-512(s, $PMK_{\text{dictionary word}}$)
- Compute u = SHA-256(A, B), k = SHA-512(N,g)

- Calculate $SK_{generated} = (B- kg^{x})^{a+ux}$

  OR

  $SK_{generated} = (Av^{u})^{b}$
- Calculate $IC1_{generated}$ = SHA-512(s, B, SHA-512($SK_{generated}$))
- Verify $IC1_{generated}$ = IC1 (obtained from traffic analysis) , if not repeat with different $PKM_{dictionary\ word}$

Theorem Statement: "EAP-SRP-PSK is resistant to dictionary attack"

Proof:

Though an adversary can obtain values N, g, A, B, s, IC1 and IC2 as shown in figure 4.1, these steps can be performed only if the adversary has knowledge of either of the private keys „a" or „b".Step 3 of the above procedure require knowledge of either a or b to generate session key SK. Since a and b are random 128 bit private keys chosen by the client and access point respectively and since at no point of the protocol are the private keys exposed, the attacker cannot perform an offline dictionary attack on the protocol as explained above. However, there are two ways by which the attacker can try to obtain the private keys.

Method 1:

If an adversary has infinite computing power and time, he/she can try all possible private keys with every dictionary word. Since private keys a and b are each of size 128 bits, the adversary has to try 2^{128} possible private keys for each dictionary word. Hence, for a dictionary with „n" words, the adversary would have to compute $[(n * 2^{128}) – 1]$ comparisons in worst case.

Performing $[(n * 2^{128}) – 1]$ computations in a reasonable time is believed to be technically infeasible (Williams, n.d) for an adversary. Hence this approach is not a feasible and cannot be used for performing a feasible dictionary attack on the protocol.

Method 2:

The adversary can use Discrete Logarithm to compute either of private keys a or b. i.e. calculate:

$a = \log_g A \bmod N$

OR

$b = \log_g B \bmod N$

However, N is a 1024 bit number and discrete logarithm of such a large number is a believed to be a hard problem (Gordon, 1993). Hence computing the private key using discrete logarithm by an adversary in a reasonable time is believed to be infeasible.

As both of the methods explained above are believed to be infeasible for an adversary to obtain the private keys, the protocol is secure against offline dictionary attacks, Proved.

CHAPTER 5. CONCLUSION

EAP-SRP-PSK provides a secure standard for wireless authentication in home wireless networks and small business networks. The protocol is resistant to dictionary attacks, Man-In-The-Middle attacks and provides random and unique session keys that help secure wireless communication within the networks.

5.1. Discussion

EAP-SRP-PSK protocol is ideal for small wireless networks with low infrastructure costs, as the protocol uses pre-shared key for authentication, without support of PKI or transport layer security. Since this protocol eliminates the need for PKI and the requirement to obtain certificates from a trusted third party, it results in overall cost reduction during installation and maintenance of networks implementing this protocol. However, since the protocol uses large prime numbers and performs large calculations, it is possible that the requirement for memory and storage space of the wireless routers may increase, which could in turn result in increased production cost of the router. Performance benchmark and protocol implementation are beyond the scope of this thesis and hence cannot be used to determine the overhead costs associated with the implementation of this protocol at this point. Hence the question “Is it possible to implement EAP-SRP-PSK in a cost efficient manner that is cheaper than or equal to the cost of current implementations of WPA and WEP?” still remains open for further research and analysis.

Another cost effective feature of this protocol is the ability to provide authentication without the support of directory services or radius servers. It is possible for a home user or small scale network to adopt any of the commercially available wireless solutions or protocols that use directory services or radius servers for authentication. However, these enterprise protocols and the infrastructure required to sustain them cannot be setup by a person without prior technical knowledge of these technologies or wireless protocols. EAP-SRP-PSK, being a pre-shared key implementation of SRP protocol, does not require any directory service or radius server and answers the question that was raised as part of significance of this research, “Can we design a protocol, without using PKI or complicated infrastructure, that can be implemented in an off the shelf wireless router and can be installed and used by a person without prior technical training or wireless technology background?”. An end user with the knowledge of a pre-shared key can ideally configure a wireless router implementing EAP-SRP-PSK without any other technological overheads. Hence this protocol can be adopted and used by users without specialized training or technological background.

EAP-SRP-PSK protocol still needs to be analyzed for backward compatibility with routers implementing currently available protocols like WPA and WEP. Firmware upgrades could ideally be used for implementing this protocol as a software based solution. However, the protocol also needs to be analyzed for backward compatibility with current hardware‟s implementing existing solutions like WPA and WEP. The possible requirement for increased memory and storage space could however be a limiting factor that could affect backward hardware compatibility.

The researcher believes that EAP-SRP-PSK, if implementable as an affordable psk based solution to replace WPA and WEP, would provide a protocol which is resistant to dictionary attack, provides security without PKI and can be configured and used by end users without specialized training or wireless technology background.

5.2. Future Work

Based on the discussions and analysis in section 5.1, the researcher aims at looking more into protocol implementation in the future. Protocol implementation and performance analysis would provide more insight into hardware requirements of the protocol, which would in turn provide an estimate of cost effectiveness of the implementation.

Future work also encompasses analysis of backward compatibility of the protocol with respect to software implementation and hardware requirements. Data fragmentation, reassembly and packet structure are also some of the areas the researcher looks forward to working in future.

Future work would also emphasize on the design of a robust and secure packet sequence and counter mechanism that can ensure secure data transmission, avoid replay attacks and denial of service attacks.

REFERENCES

REFERENCES

Aircrack-ng (n.d.) – In aircrack-ng.org .. Retrieved September 20 2009, from http://aircrack-ng.org.

Allen, J., & Wilson, J. (2002). Securing a wireless Network. Proceedings of the 30th annual ACM SIGUCCS conference on User services, 213-215. doi: http://doi.acm.org/10.1145/588646.588696.

Beck, M. & Tews, E. (2008). Practical attacks againstWEP and WPA. Conference On Wireless Network Security. Doi: http://doi.acm.org/10.1145/1514274.1514286.

Borisov, N., GoldBerg, I. & Wagner, D. (2001). Intercepting Mobile Communications: The Insecurity of 802.11. Proceedings of the 7th annual international conference on Mobile computing and networking, 180-189. doi: http://doi.acm.org/10.1145/381677.381695.

Carlson, J., Aboba, B. & Haverinen, H. (2002). EAP SRP-SHA1 Authentication Protocol. Retrieved from http://tools.ietf.org/html/draft-ietf-pppext-eap-srp-03

Congdon, P., Aboba, B., Zorn, G. & Roese, J. (2003). RFC 3580: IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines. Retrieved from http://www.rfc-archive.org/getrfc.php?rfc=3580.

coWPAtty (n.d.) – In wirelessdefense.org.. Retrieved September 20 2009, from http://wirelessdefence.org/Contents/coWPAttyMain.htm.

Dictionary Attack (n.d.). Retrieved March 3, 2010, from http://www.rsa.com/products/bsafe/documentation/cryptoc62html/group__AD__CMMN__GLOSSARY.html.

Diffie-Hellman key exchange (n.d). Retrieved March 3, 2010 from http://www.netip.com/articles/keith/diffie-helman.htm.

Gordon, D. (1993). Discrete Logarithms in GF(p) via the number field sieve. SIAM Journal on Discrete Mathematics, 124-138. Doi: 10.1137/0406010

Han, L. (2006). A Threat Analysis of the Extensible Authentication Protocol. Retrieved from www.scs.carleton.ca/~barbeau/Honours/index.html.

IEEE 802.11 – Wikipedia, the free encyclopedia. (n.d). Retrieved December 6, 2009, from http://en.wikipedia.org/wiki/802.11.

IEEE 802.11i-2004 (2004): Amendment 6: Medium Access Control (MAC) Security Enhancements. Retrieved from http://standards.ieee.org/getieee802/download/802.11i-2004.pdf.

Lehembre G. (2005). Wi-Fi security – WEP, WPA and WPA2. Proceedings of the 7th annual international conference on Mobile computing and networking, Retrieved from http:// www.hsc.fr/ressources/articles/hakin9_wifi/index.html.en.

Manganaro, A., Koblensky, M., & Loreti, M. (2009). Design of a Password-Based EAP Method. E-business and Telecommunications (Vol. 23). Springer Berlin Heidelberg. Doi: 10.1007/978-3-540-88653-2

Rantwijk, J.V. (n.d), WPA key calculation: From passphrase to hex. Retrieved September 21, 2009, from http://www.xs4all.nl/~rjoris/wpapsk.html.

Schneier, B. (n.d). Man-In-The-Middle-Attack. Retrieved March 3, 2010 from http://www.schneier.com/crypto-gram-0404.html#6.

SRP Design Specification. Retrieved January 28th, 2010 from http://srp.stanford.edu/design.html

Williams, C.L. (n.d), “A Discussion of the Importance of Key Length in Symmetric and Asymmetric Cryptography”, Retrieved April 15, 2010 from http://www.giac.org/certified_professionals/practicals/gsec/0848.php

Wright, J. (n.d.). ASLEAP ReadMe. Retrieved September 20, 2009 from http://asleap.sourceforge.net/README.

Zorn, G., "RFC 2759 - Microsoft PPP CHAP Extensions, Version 2. (n.d.)", Retrieved September 20, 2009 from http://tools.ietf.org/html/rfc2759#section-3.