802.11 무선패킷 전송을 위한 새로운 Rekeying 스킴

팽상우* · 태유슈* · 치시안양* · 이훈재** · 임효택**

*동서대학교 유비쿼터스IT학과

A New Rekeying Scheme for 802.11 Wireless Packets Transmission

Seong-Yee Phang* · Yu-Shu They* · Chi-Shian Yang* · HoonJae Lee** · Hyotaek Lim**

*Department of Ubiquitous and IT

Graduate School of Design and IT, Dongseo University

Busan, 617-716, South Korea

E-mail : {seongyee,ysthey,csyang}@dit.dongseo.ac.kr, {hjlee,htlim}@dongseo.ac.kr

ABSTRACT

Rekeying is the process of changing the encryption key of an ongoing communication. The main objective is to limit the amount of data encrypted with the same key. The IEEE 802.11 standard defines the Wired Equivalent Privacy, or WEP, encapsulation of 802.11 data frames. MAC at sender encrypts the payload (frame body and CRC) of each 802.11 frame before transmission using RC4 stream cipher. MAC at receiver decrypts and passes data to higher level protocol. WEP uses symmetric key stream cipher (RC4) where same key will be used for data encryption and decryption at the sender and the receiver. WEP is not promising with the advancement of the wireless technology existing today. We propose to use the existing information to define the security attributes. This will eliminate the steps that regenerated keys have to be sent to each other over certain period. The rekeying scheme is according to the number of bytes transmitted. Therefore, even the attacker has recorded the packets, it will be insufficient information and time for the attacker to launch the attacks as the key is not deterministic. We develop a packet simulation software for packet transmission and simulate our propose scheme. From the simulation, our propose scheme will overcome the weak WEP key attack and provide an alternative solution to wireless packet transmission. Besides that, our solution appears to be a software approach where only driver updates are needed for the wireless client and server.

키워드

WEP, Rekeying, Wireless LAN, Encryption

I . Introduction

Rekeying is the process of changing the encryption key of an ongoing communication. The main objective is to limit the amount of data encrypted with the same key. In contemporary systems, rekeying is implemented by forcing a new key exchange such as in Internet Key Exchange (IKE) [1]. In a securecommunication where A Trust B, B Trust A (A->B, B<-A), both parties hold same information. The IEEE 802.11 standard defines the Wired Equivalent Privacy, or WEP [2], encapsulation of 802.11 data frames. MAC at sender encrypts the payload (frame body and CRC) of each 802.11 frame before transmission using RC4 stream cipher [3]. MAC at receiver decrypts and passes data to higher level protocol. WEP uses symmetric key stream cipher (RC4) where same key will be used for data encryption and decryption at the sender and the

receiver. WEP is not promising with the advancement of the wireless technology, WLAN existing today [4],[5].

II. Motivation and Objectives

We assume that a laptop is connecting to a 802.11g network and transferring data at 54mbps with no packet loss issue and full bandwidth guaranteed. Therefore, approximately 4718 packets will be transferred per second and we assume each packet is at maximum size of 1500 bytes. An attacker who has recorded the traffic can launch the well known Birthday Attack [6] to get the WEP key with 99% of collision that happened in only 3 seconds. We propose to use the existing information to define the security attributes. This will eliminate the steps that regenerated keys have to be sent to each other over certain period. The rekeying scheme is according to the number of bytes transmitted. Therefore, even the attacker has recorded the packets, it will be insufficient information and time for the attacker to launch the attacks as the key is not deterministic. This appears as our main objective of this research work. Besides, our solution can be known as a software approach where only driver updates to both the client and server to provide a better security.

III. Propose Solution

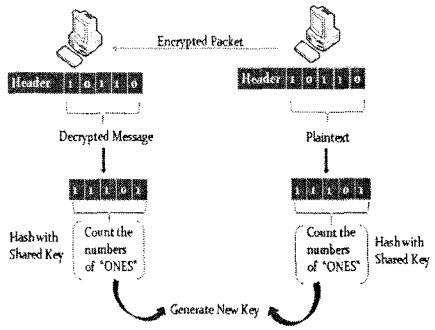

Figure 1. Overview.

As illustrated in figure 1, this is the overview of our proposed solution. This figure shows two computer which are the communicating parties. Let's assume one party as the wireless access point where acts as a third person who securely communicate with another wireless client through wireless channel. First of all, the client will encrypt the packets payload with the pre-shared key and the default initialization vector. The server will decrypt the message accordingly with the same pre-shared key and initialization vector. For the next packet transmission, the client will generate a new encryption key which is the result of the hash function by counting the numbers of 1 of the data in raw form with the pre-shared key. For the server, it will generate the same decryption key after receiving the data packet from the client by doing the same algorithm as in the client side.

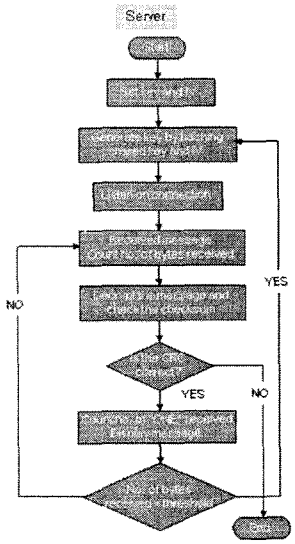

Figure 2. Server Flowchart.

As illustrated in figure 2, this will be the flowchart of our proposed solution in the server side. First, the user will key in the pre-shared key and the initialization vector to the server side. By hashing the shared key and the initialization vector, the key used in the communication will be produced. The server will then start receiving the message and count the number of bytes being received. The message will be decrypted using the key described earlier on and the checksum of the message will be validated. If the checksum is correct, the no of "ONE" received will be counted and the message will be displayed accordingly which indicates a successful communication with the client. If the number of bytes count more than the threshold values, the generation of a new key will be repeated.

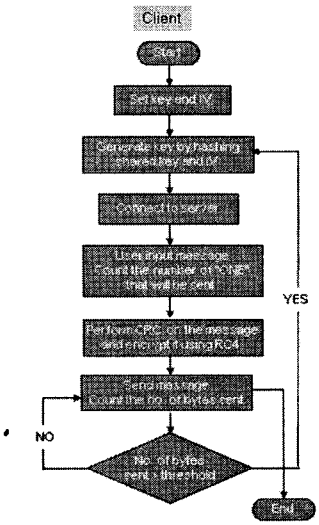

Figure 3. Client Flowchart.

As illustrated in figure 3, this will be the flowchart of our proposed solution in the client side. First, the user will key in the pre-shared key and the initialization vector to the client side. By hashing the shared key and the initialization vector, the key used in the communication will be produced. The client will then connect to the server and send the message. The number of "ONE" which contains in the packet will be counted. The message will be encrypted using the key described earlier on after the checksum be calculated. The encrypted message will be send accordingly and if the number of bytes count more than the threshold values, the generation of a new key will be repeated.

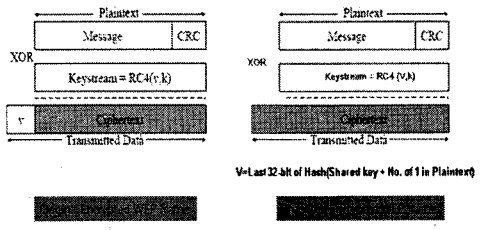

Figure 4. Frames Format.

Figure 4 illustrates the default wep frame format and the frame format of the proposed solution. The initialization here will be replaced by the last 32 bit of hash function of the pre-shared key and the number of "ONE" counted in the plaintext. The algorithm can be concluded by the representation in figure 5 below.

Initial:

C_0=IV

K_0=Master Key

Iteration T:

$K_T=H(K_{T-1},C_{T-1})$

$C_T=L(K_T)$

$C_T=C_T+F(P_T)$

Figure 5. Algorithm.

The notation of the algorithm is as follow:- P is the plaintext, Key is the key, H is the 128 bit Hash fucntion, T is the period within rekeying, C is the bit counter of 1 in plaintext, F will the function to count the number of 1 in the plaintext and L is the function to get the last 32 bit value from the 128 bit key.

IV. Simulation and Results

We develop a packet simulation software for packet transmission and simulate our propose scheme. We time the packets with the assumption in 802.11g network, our scheme will rekey before 357913 packets where approximately at each 75 seconds. This indicated that our propose scheme will overcome the weak WEP key attack and provide an alternative solution to wireless packet transmission. Besides that, our solution appears to be a software approach where only driver updates are needed for the wireless client and server.

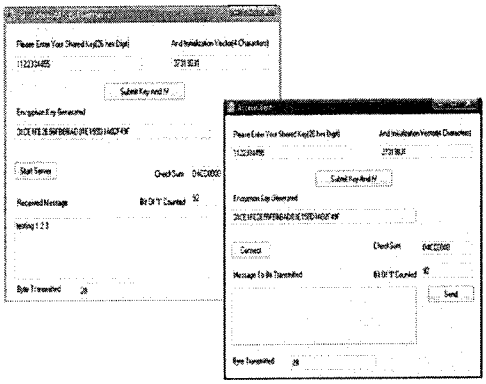

Figure 6. First Simulation.

Figure 6 illustrates the initialization of the simulation. From here, we show that the same pre-shared key and initialization hold at both side of the server and the client.

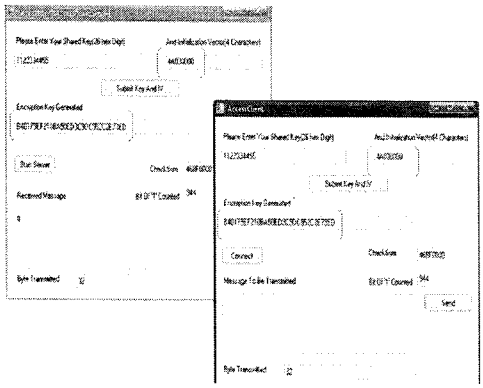

Figure 7. Second Simulation.

Figure 7 illustrates the initialization vector changing according to the transferred "One" in the plaintext and the encryption key changing as well. Both the client and the server still hold the same key after bytes of communication.

VI. Conclusion

As a conclusion, we proposed a new rekeyinh scheme for 802.11 wireless packets transmission. The proposed solution has been simulated as a prove of concept. The estimation from the simulation proved that the packets with the assumption in 802.11g network, our scheme will rekey before 357913 packets where approximately at each 75 seconds. This indicated that our propose scheme will overcome the WEP key attack and provide an alternative solution to wireless packet transmission. Besides that, our solution appears to be a software approach where only driver updates are needed for the wireless client and server. However, more work need to be done in future for further evaluating the proposed solution by using well known security test. Besides that, implementation on the real access point is needed for performance evaluation.

References

[1] Harkins D,Carrel D.The Internet key exchange (IKE).RFC 2409,1998.

[2] A. Stubblefield, J. Ioannidis, A. D. Rubin. A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP).ACM Transactions on Information and Systems Security,2004,7(2): 319-332.

[3] B. Schneier, *Applied Cryptography*, 2nd edition ed. New York:Wiley, 1996.

[4] BertinPhilippe,Lebeugle Franck, Journe Thierry. WLAN standards and evolution. Annales des Telecommunications, 2003,58(3-4):337-368.

[5] Majstor, F., WLAN security threats & solutions, Local Computer Networks, 2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on , 20-24 Oct. 2003.

[6] Menezes, A.J, Van Oorschot, P.C, Vanstone, S.A, "Handbook of applied cryptography", 1997.