11-2005

Anonymous DoS-Resistant Access Control Protocol using Passwords for Wireless Networks

Zhiguo WAN
National University of Singapore

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Feng BAO
Institute for Infocomm Research, Singapore

Akkihebbal L. ANANDA
National University of Singapore

Citation

WAN, Zhiguo; DENG, Robert H.; BAO, Feng; and ANANDA, Akkihebbal L.. Anonymous DoS-Resistant Access Control Protocol using Passwords for Wireless Networks. (2005). *IEEE Conference on Local Computer Networks 30th Anniversary: Proceedings: Sydney, Australia, November 15-17, 2005*. 328-335.
Available at: https://ink.library.smu.edu.sg/sis_research/523

Anonymous DoS-Resistant Access Control Protocol Using Passwords for Wireless Networks

Zhiguo Wan*†, Robert H. Deng‡, Feng Bao† and Akkihebbal L. Ananda*
* School of Computing, National University of Singapore, Singapore 117543
{wanzhigu,ananda}@comp.nus.edu.sg
† Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
{zhiguo,baofeng}@i2r.a-star.edu.sg
‡ School of Information Systems, Singapore Management University, Singapore 259756
robertdeng@smu.edu.sg

Abstract— **Wireless networks have gained overwhelming popularity over their wired counterpart due to their great flexibility and convenience, but access control of wireless networks has been a serious problem because of the open medium. Passwords remain the most popular way for access control as well as authentication and key exchange. But existing password-based access control protocols are not satisfactory in that they do not provide DoS-resistance or anonymity. In this paper we analyze the weaknesses of an access control protocol using passwords for wireless networks in IEEE LCN 2001, and propose a different access control protocol using passwords for wireless networks. Our new protocol avoids the weaknesses of the previous protocol, and the client can anonymously authenticate himself to the server with a human-memorable password, while the server is free of DoS attacks. We also present detailed security and performance analysis for our protocols, and show that our protocol is both secure and efficient for access control in wireless networks.**

Wireless Networks, Security Protocol, Access Control

I. Introduction

Because of great convenience and flexibility provided by wireless networks, the popularity of wireless networks has surged dramatically over the recent few years. This also leads to the increasing pervasiveness of wireless technologies, such as IEEE 802.11, HomeRF, HIPERLAN/2 and Bluetooth. Though wireless networks offer great benefits, they are more susceptible to attacks and require more protection than their wired counterpart. In wireless networks, data is broadcast in the open air, and it is impossible to have physical controls over the transmission boundaries. That makes eavesdropping or active attacks more easily than in wired networks, and hence security becomes the major concern in wireless networking. Since deployment of wireless network technologies in public places bears the danger of unauthorized users gaining access to network services, it is extremely crucial to be able to restrict access to the network only to authorized users. Therefore, secure user authentication and authorization, and a reliable access control mechanism are vital for wireless networks.

Human-memorable passwords have been widely used for authentication and key exchange due to their user friendliness. Hence it is desirable to use shared passwords to achieve access control in wireless networks. The authentication and key exchange protocols that use weak passwords, known as the password-authenticated key exchange (PAKE) protocols, have been well investigated in the literature. However, existing PAKE protocols fail to provide client identity confidentiality and DoS-resistance for wireless networks.

In wireless environment, there is an important requirement on confidentiality protection of a client's identity, which has been neglected in many solutions for wireless networks. In wireless networks, the current location and the movement of a roaming user are important parts of the user's privacy, and they should be protected during communications. Knowing the user's identity helps the attacker to locate the user and track his movement, so it is important for a protocol to provide identity confidentiality to users in a wireless environment. In PAKE protocols, the client needs to disclose its identity so that the server knows which password is used for authentication and key exchange. Consequently, client identity confidentiality cannot be achieved with PAKE protocols.

Denial-of-service (DoS) attack is a serious threat against availability of network services, and it is exceedingly difficult to counter against such attacks in wireless networks. Wireless networks are extremely vulnerable to DoS attacks because of its unique characteristics. In wireless networks, both passive attacks and active attacks can be launched easily since data transmission happens in the open air. Moreover, wireless networks usually has limited bandwidth, power and computation capability, and hence they are more susceptible to DoS attacks. But PAKE protocols have no built-in mechanism to counter against DoS attacks. In PAKE protocols, the server can only authenticate the client after expensive computation, which results in vulnerability against DoS attacks.

In this paper, we review an access control protocol proposed called Lancaster protocol in IEEE LCN 2001 [22], and analyze its weaknesses and design flaws. Besides the password shared between the server and each client, this protocol additionally requires the server has a certificate issued by a well-known authority. Under the same setting, Then we propose a new protocol using human-memorable passwords for access control in wireless networks. Our protocol avoids the weaknesses and design flaws of the Lancaster protocol, and offers two more important features for wireless networks : client identity confidentiality and resistance against DoS attacks.

Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)
http://dx.doi.org/10.1109/LCN.2005.30

The remainder of this paper is organized as follows. In Section II, we first review related work on security protocols for wireless networks; we then analyze the Lancaster protocol, and present a new protocol that can achieve DoS-resistance and identity confidentiality for wireless networks in section III. After that, we discuss and analyze the security and performance of our protocol. Finally, we draw our concluding remarks in Section IV.

For ease of reference, important notations used throughout the paper are listed in the following table I:

TABLE I
NOTATION

C	The client
S	The authentication server
$E_X(M)$	Message M encrypted with X's public key
$e_K(M)$	Symmetric key encryption of a message M using K as the encryption key
$H_k(\cdot)$	Cryptographically secure one-way keyed hash function

II. Related Work

There have been a lot of research efforts on access control and authentication protocols, and some of them are specially designed for wireless networks. Unfortunately, they either cannot fulfill all the security requirements of wireless networks, or need each client has his own PKI certificate, which is too heavy a burden for most organizations.

The IEEE standard 802.11 [17] has used the Wired Equivalent Privacy (WEP) protocol, which is a symmetric cryptosystem based protocol, for access control in wireless networks. WEP is intended to protect wireless communication from eavesdropping as well as preventing unauthorized access to wireless networks. It replies on a shared secret between the mobile station and the access point to achieve the aforementioned goals. However, it has been indicated that WEP has serious design flaws that make WEP vulnerable against both passive and active attacks [2], [6].

To solve the above security problems, IEEE specifies the 802.11i standard [19] to enhance the security of 802.11. In the 802.11i standard, a long term security architecture for 802.11 called the Robust Security Network (RSN) and the Robust Security Network Association (RSNA) are defined for wireless networks. RSNA uses the IEEE 802.1X standard [18] to enhance access control, authentication, key management, and key establishment mechanisms for 802.11. In IEEE 802.1X standard, EAP (Extensible Authentication Protocol) is used for authentication and key establishment. Although it has been pointed out that the 802.1X protocol is vulnerable to the session hijacking attack and the man-in-the-middle attack [16], these problems do not exist when security protocols that provide strong mutual authentication over EAP is used between access points and mobile stations.

EAP, which is defined in RFC 2284, is a flexible protocol used to carry arbitrary authentication information. It provides flexible and extensibility for authentication by defining an independent message exchange layer. A set of IETF drafts have defined different security protocols over EAP: EAP-TLS, EAP-TTLS (Tunneled TLS), PEAP (Protected EAP), LEAP (Lighweight EAP) etc. However, each protocol of them has its own drawbacks. LEAP is an enhanced version from EAP-MD5 which is insecure against dictionary attacks, which results in insecurity of LEAP against dictionary attacks too. EAP-TLS replies on certificates on both the server and the client side to deliver mutual authentication and secure key exchange, which is too great a barrier for most organizations to overcome. EAP-TTLS and PEAP are developed to overcome the PKI barrier in EAP-TLS. Both protocols only require the server certificate to establish a TLS tunnel in stage one, and then authenticate each other in stage two. But they are susceptible to DoS attacks because the server requires to compute a signature on authentication request of any entity. Moreover, client identity confidentiality is not provided in EAP-TLS, LEAP and PEAP.

Several EAP methods using weak passwords also have been proposed as IETF drafts, such as EAP-PAX, EAP-SRP, and EAP-SPEKE. The protocols using only weak passwords for authentication and key exchange, known as the PAKE protocols, have gained extensive attention and research interests until now. The IEEE P1363 Standard Working Group is engaged in standardization on password-based public-key cryptographic protocols, including SPEKE [12], SRP [25], PAK [20] and AMP [15]. The main disadvantages of pure PAKE protocols are their incapability of client identity protection and susceptibility against DoS attacks, and hence they are not suitable for access control in wireless networks. In PAKE protocols, the client requires to disclose his identity to the server so that the server knows which password should be used for authentication. As a result, such protocols cannot provide identity confidentiality for clients. On the other hand, in such protocols the server can only authenticate the client after expensive computation. This causes the protocols susceptible to DoS attacks, since anyone can send requests to launch the server into computational expensive operations.

Unlike the pure PAKE protocols, the EAP-PAX protocol can provide client identity confidentiality when the server holds a certificate, which is the same with our protocol. However, it has several design flaws and cannot meet all requirements of wireless networks. First of all, it is vulnerable to dictionary attacks during its registration phase if the server does not have a certificate. Besides, the protocol replaces the weak password on both the server and the client side with a generated random secret on each update. As a result, the protocol doesn't obtain convenience of using human-memorable passwords in later authentication. Furthermore, the protocol is susceptible to DoS attacks since any part can trick the server into expensive public key cryptographic decryption.

Many other public key cryptosystem based protocols have also been proposed for authentication in wireless networks, but they usually fall short of one or more requirements. The Beller-

Chang-Yacobi protocol [4] and the Aziz-Diffie protocol [3] do not provide client identity confidentiality, while the Boyd-Park protocol [9] do not obtain perfect forward secrecy. Though the ASPeCT protocol provides perfect forward secrecy and client identity confidentiality, but it is susceptible to DoS attacks. The Just Fast Keying (JFK) protocol [11] specified in the Internet draft provides immunity to DoS attacks and identity confidentiality for clients, but it requires certificates on the client end which is too big a hurdle to deal with. Another Internet draft specified the Internet Key Exchange protocol (IKEv2) [13] also has such a PKI barrier problem that makes it undesirable to be used in wireless networks.

III. Access Control Protocols for Wireless Networks

In this section, we first review the Lancaster access control protocol proposed in IEEE LCN 2001 [22], [10] and discuss its security weaknesses. Then we introduce a new access control protocol for access control in wireless networks. Both the Lancaster protocol and our protocol has the same system setup: a client shares a password with a server, while the server has a certificate issued by a well-known authority.

The authentication of WLAN is based on a 3-party model as shown in Fig. 1: the client, which requires access to the WLAN; the access router, which grants access to the client; and the authentication server, which authenticates the client and gives permission to the client. If the client intends to access the wireless network, he requires to mutually authenticate with the authentication server and agrees on a session key with the server. After that, the session key is transmitted to the access router for the purpose of access control.

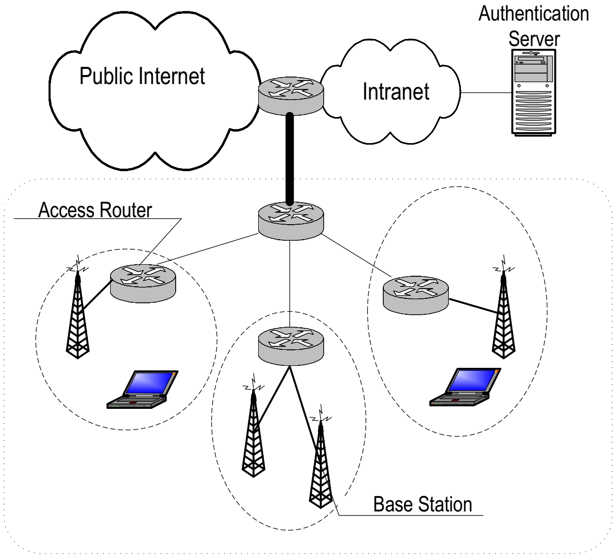

Fig. 1. Overview of the Wireless Networks

A. System Settings and Security Requirements

The system settings of the protocols are as follows: the authentication server and the client share a *weak* (low-entropy) password, and such a password is susceptible to dictionary attacks. Meanwhile, the authentication server has a certificate issued by a well-known authority. We assume that the adversary has complete control of the wireless network, and he has reasonable computation capability. He can eavesdrop, drop, modify, inject, delay and replay messages transmitted over the wireless link.

We list below a number of security requirements for such protocols.

- *Mutual Authentication:* Authentication of the client to the authentication server and authentication of the server to the client. The network want to be sure that it is communicating with a genuine client; otherwise there is a danger that spurious client will be able to fraudulently gain a level of service without ever intending to pay for the service. Authentication of the authentication server to the client is also necessary in order to prevent a type of man-in-the-middle attack as described in [16].
- *Key Authentication and Key Confirmation:* Key authentication requires that only the legitimate participants in the protocol but no other entity possess the agreed secret key, while key confirmation means that both parties in the protocol can be assured that both of them derive the same secret key.
- *Perfect Forward Secrecy:* Previous session keys and confidential messages should be protected against compromise of the passwords and other long-term secrets.
- *Secure Against Dictionary Attacks:* Since passwords must be memorable, a secure password based protocol should resist brute-force guessing, or dictionary attacks.
- *Client Identity Confidentiality:* Confidentiality protection of a client's identity against both passive and active attacks. In the wireless environment, the current location and the movement of a roaming user are important parts of the user's privacy, and they should be protected during communications. Knowing the user's identity helps the attacker to locate the user and track his movement, so it is important for a protocol in a wireless environment to provide identity confidentiality to users.
- *Protection Against DoS Attacks:* The protocol should have certain built-in remedies to reduce the effect of DoS attacks aiming to exhaust the server's computation resource (computation-DoS) or storage resource (memory-DoS).
- *Access Control:* Only authorized clients can obtain access to the wireless network. To protect from the parking lot attacks [2], fine grained access control, ideally on a per packet level, should be enforced.

B. The Lancaster Access Control Protocol

The Lancaster access control architecture [22], [10] is for publicly accessible wireless overlay networks. It is designed to address the problem of ubiquitous Internet service provisioning within the city of Lancaster. As illustrated in Fig. 2, it consists of three messages.

1) To access the network, a client initiates the process by sending an authentication request to the authentication server via an access router:

$$E_S(MAC_C, IP_C, K, Username, Password) \tag{A.1}$$

where MAC_C, IP_C, $Username$ and $Password$ are the client's MAC address, IP address, username and password, respectively, and K is a secret session key generated by the client. This request is encrypted with the public key of the authentication server.

2) Upon receiving the authentication request, the authentication server decrypts it using its private key. It checks the received password with the one in its database. If the two passwords match, the client is considered authentic. The authentication server then generates an authentication token $Token$, encrypts it using the session key K and sends the ciphertext to the client:

$$e_K(Token) \tag{A.2}$$

3) Next, the authentication server encrypts the client's MAC address, IP address, the access token and the session key with the access router's public key, and sends the result to the access router:

$$E_{AR}(K, Token, MAC_C, IP_C) \tag{A.3}$$

The access router decrypts this message and stores MAC_C, IP_C, $Token$ and K into an access control list (ACL).

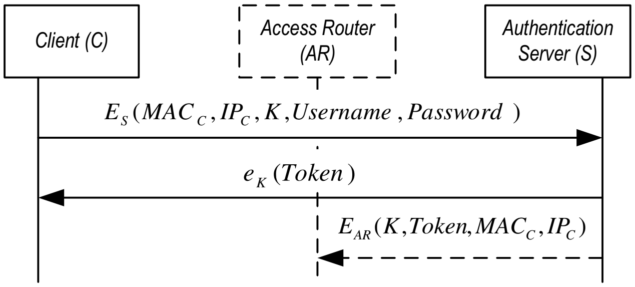
Client (C)
Access Router (AR)
Authentication Server (S)
$E_S(MAC_C, IP_C, K, Username, Password)$
$e_K(Token)$
$E_{AR}(K, Token, MAC_C, IP_C)$

Fig. 2. The Lancaster Protocol.

When the client sends a packet to the network, it includes an access control extension header in the IP packet as illustrated in Fig. 3. This header contains the access token and a checksum both encrypted with the session key K using a symmetric key cipher. In Fig. 3, V denotes the protocol version, T denotes the type of services, and Res denotes the reserved bits.

The access router checks packets from the clients (i.e., the wireless network) for purpose of access control. When a packet is received from the wireless network, the access router looks up the MAC address in the ACL. If an entry for the client device exists, the access router verifies the IP source address. In the case of a match, it decrypts the access token and the checksum using the session key and validates its content against the ACL. When successful, the access control extension header is stripped off and the packet is passed on. Packets that fails any of those tests are dropped. One exception to this rule is that when a client is first seen in a cell, it is allowed to contact certain well-known IP addresses; this allows clients to initially communicate with the authentication server.

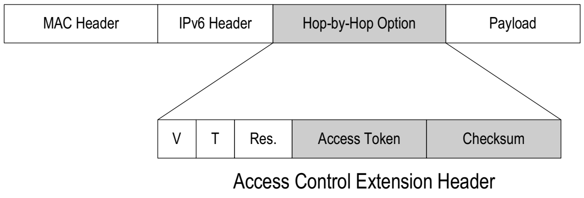

Fig. 3. The Packet Header Format in the Lancaster Protocol.

The Lancaster protocol is simple in design, with only two message exchanges between a client and the authentication server and one message sent from the authentication server to an access router. Unfortunately, it has many serious security flaws which make it vulnerable to various attacks.

First of all, the protocol does not follow the well-known "challenge-response" principle. Specifically, message (A.1) is not sent in response to any challenge from the authentication server. This makes the protocol subject to the replay attacks. Obviously, message (A.1) can be replayed by anyone to the authentication server and the server will accept the message and believes the sender as authentic.

Secondly, if the attacker is able to compromise just one session key K, he can gain full access to the wireless network. The attacker can replay message (A.1) to request authentication, and then with the knowledge of the session key the attacker can obtain the access token by decrypting message (A.2). Employing techniques of IP spoofing and MAC spoofing, the attacker can then gain full access to the wireless network with the access token. The attacker can perform this attack any time he wishes to with the knowledge of just one session key even he does not know the client password at all.

Thirdly, since the password space is normally small, the attacker can perform dictionary attacks against message (A.1) if any session key is exposed to the attacker. And this attack can disclose the client's password.

Finally, the protocol does not provide key confirmation for both parties. Neither party is ensured that the other party shares the same secret session key. In message (A.2), $Token$ is selected and encrypted with K by the server, hence the client is unable to confirm that they share the same session key K. Also, key freshness is not guaranteed in the protocol. There is no mechanism to prevent reuse of old session keys. If the client reuses a previously used session key, the server will simply accept this old key. This leads to the failure of the protocol when only one session key is compromised.

Moreover, the technical details of the access control extension header shown in Fig. 3 is not clearly spelled out in [22], [10]. Since the client and the access router share a secret session key K, the use of the access token is not clear. We note that the combined use of checksum and symmetric key encryption as in the Lancaster protocol is dangerous if not designed carefully [5], [6]. The description on the construction of the access control extension header in [22], [10] does not provide enough technical details for us to make creditable analysis.

C. Our Access Control Protocol

As discussed earlier, the intrinsic characteristics of PAKE protocols lead to their incapability of providing client identity protection and susceptibility against DoS attacks. In this section, we propose a secure protocol to meet all the requirements.

Before the protocol starts, the client and the server agree on a set of security parameters: a multiplicative group $\mathbb{Z}_p^*$, its subgroup $\mathbb{G}_{p,q}$ of order q and a generator g of $\mathbb{G}_{p,q}$, where p, q are large prime numbers. Specifically, p is selected as a safe prime or a secure prime, which means that $p = 2q+1$ or $p = 2qr+1$ where all the factors of r are comparable to q. Assuming that discrete logarithm problem over $\mathbb{G}_{p,q}$ is hard.

Before a client can access network services, it performs the following authentication and key agreement with the authentication server. The protocol message exchanges are illustrated in Fig 4.

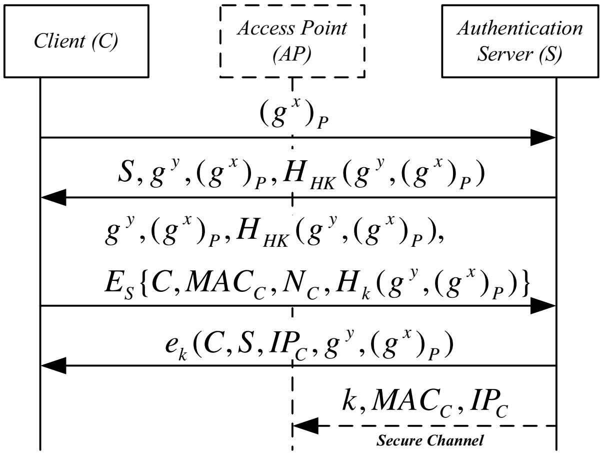
Client (C)
Access Point (AP)
Authentication Server (S)
$(g^x)_P$
$S, g^y, (g^x)_P, H_{HK}(g^y, (g^x)_P)$
$g^y, (g^x)_P, H_{HK}(g^y, (g^x)_P),$
$E_S\{C, MAC_C, N_C, H_k(g^y, (g^x)_P)\}$
$e_k(C, S, IP_C, g^y, (g^x)_P)$
k, MAC_C, IP_C
Secure Channel

Fig. 4. Our Anonymous DoS-Resistant Access Control Protocol.

1) The client C chooses a random number $x \in_R Z_q$ and computes the exponential $(g^x)_P$ encrypted with its password. Then he sends

$$(g^x)_P \tag{B.1}$$

to the server, where C is the identity of the client. Since the exponential g^x is randomly generated, dictionary attacks are not applicable to disclose the client's password.

2) After the server receives the first message, he chooses a random number $y \in_R Z_q$ and the exponential g^y. He computes a hash $H_{HK}(g^y, (g^x)_P)$ with a hash key HK private to the server only. Then the server sends

$$S, g^y, (g^x)_P, H_{HK}(g^y, (g^x)_P) \tag{B.2}$$

to the client.

In this step, the server should avoid expensive computation in order to resist DoS attacks, because the server cannot determine whether the client is valid. Actually the computation cost of the server includes an exponentiation and a hash computation in this message. But the server can generate the random exponential beforehand or periodically, so that the computation cost of the server is only a light-weight hash computation. In this message, HK is a hash key known only by the server, and it is updated frequently to prevent accidental disclosure. As a result, the hash $H_{HK}(g^y, (g^x)_P)$ can serve as an authenticator and a cookie that would be sent back by the client in the next message.

3) After the client receives the above message, he computes the session key $k = H(C|S|g^{xy})$ and then a hash $H_k(g^y, (g^x)_P)$. Then he fetches the server's public key to encrypt his identity C, his MAC address MAC_C, a random nonce N_C, and the hash, and then he sends

$$g^y, (g^x)_P, H_{HK}(g^y, (g^x)_P),$$
$$E_S\{C, MAC_C, N_C, H_k(g^y, (g^x)_P)\} \tag{B.3}$$

to the server.

In this step, the client derives the session key k by Diffie-Hellman computation, which provides perfect forward secrecy for the protocol. The hash $H_{HK}(g^y, (g^x)_P)$ which serves as an authenticator as well as the two exponentials is sent back to the server so that the server does not need to store the two exponentials but still can verify that the two exponentials are not modified by checking the hash. Hence the server can resist DoS attacks that intend to deplete the server's storage space. The client's identity is protected with the server's public key to avoid identity disclosure.

4) After the server receives the above message, it verifies the validity of the hash $H_{HK}(g^y, (g^x)_P)$ by looking up the hash value in its storage. If the hash value is stored on the server, then the server can verifies whether the two exponentials are valid by checking the received hash. After that, the server decrypts $E_S(C, MAC_C, N_C, H_k(g^y, (g^x)_P))$ and fetch the client's password according to his identity. The server now can decrypts $(g^x)_P$ to obtain g^x and computes the session key k in the same way as the client. At the end, the server assigns an IP address IP_C to the client and sends the following message to the client.

$$e_k(C, S, IP_C, g^y, (g^x)_P) \tag{B.4}$$

In order for access control at the access point, the server also sends the session key k, the client's MAC address and IP address to the access point through a secure channel.

After successful authentication and key agreement, the enforcement of access control in our protocol uses an access control extension header. Specifically, we follow the approach of the Authentication Header in IPSec [14]. Our access control extension header is shown in Fig.5, where the Integrity Check Value (ICV) is a keyed hash function output given by

$$ICV = H_k(MACHeader||IPv6Header|| V||T||Res||Payload).$$

Here V denotes protocol version, T denotes the type of service, and Res denotes the reserved bits.

The ICV is computed by the client C for every IP packet it sends to the network, and this provides integrity and data origin authentication for the IP packet. It can also be used to provide protection against replays by incorporating a sequence number in the extension header [14].

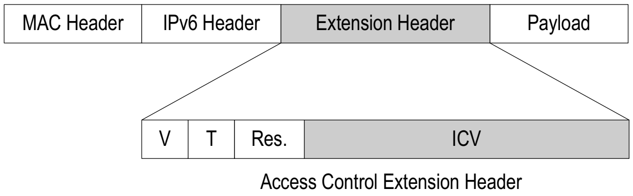

Fig. 5. The Packet Header Format in Our Protocol.

When an IP packet is received from the wireless network, the access router looks up the MAC address in the ACL. If the entry exists for the client device, the access router fetches the secret session key, computes the ICV over the appropriate fields of the received packet, using the same formula as the client, and verifies that it is the same as the ICV included in the received packet. If the verification is successful, the access control extension header containing the ICV is stripped off and the packet is passed on; otherwise, the packet is dropped silently.

D. Security Analysis of Our Protocol

In this section, we analyze the security of our protocol and show that our protocol fulfills all the requirements aforementioned, including client identity confidentiality and resistance to DoS attacks.

Mutual authentication between the client and the server is achieved after a successful protocol execution. The server authenticates the client by verifying the hash $H_k(g^y, (g^x)_P)$ encrypted with the server's public key in message (B.3), since only the legitimate client knows x and can compute the session key k. On the other hand, because only the valid server can decrypt the ciphertext in message (B.3) and know the client's identity, the client can authenticate the server by checking message (B.4). At the same time, both parties are ensured that the other party obtains the same session k as himself.

Our protocol provides perfect forward secrecy by employing Diffie-Hellman key exchange, and this ensures security of previous sessions even when the shared password or the server's private key is compromised. With the secret password P, an adversary who has stored previous communication content can decrypt $(g^x)_P$ from previous session. If the server's private key is compromised, the adversary can discover the client's identity. But in both cases, it is still computationally infeasible for the adversary to obtain k assuming the hardness of discrete logarithm. So the adversary still cannot derive the session key and in turn cannot disclose previous communication.

In our protocol, the identity of the client is protected against both passive and active attacks. After the client receives message (B.2), it fetches the public key of the server according to the server's identity, and sends its identity encrypted by the server's public key. Hence only the valid server who holds the corresponding private key can decrypt it to obtain the client's identity. Later in message (B.4), the client's identity is protected with the session key k. Since a passive attacker cannot complete Diffie-Hellman computation to derive k, he cannot obtain any information about the client's identity. On the other hand, an active attacker has no legitimate private key to decrypt the identity information in message (B.3), and he cannot complete Diffie-Hellman exchange to derive k to decrypt message (B.4) either. Therefore, both passive and active attacks cannot disclose the client's identity.

The shared password is secure against off-line dictionary attacks in our protocol. In the protocol, the password is used to encrypt the random exponential g^x generated by the client, and hence an adversary cannot verify his guess because he does not know g^x. If the adversary impersonates as the server, and sends an exponential $g^{y'}$ to the client. The client then will derive the session key $k' = H(C|S|g^{xy'})$, which can be computed by the adversary who has y'. However, the adversary cannot verify his guess by checking $E_S\{C, MAC_C, N_C, H_k(g^y, (g^x)_P)\}$ because N_C is a random nonce chosen by the client.

The server does not have to keep any state and commit any storage when sending out message (B.2), and this relieves the server from memory DoS attacks that intend to exhaust the server's memory. The two exponentials will be sent back in message (B.3), so the server does not need to create state and store these information. So if the client is fraudulent, the server will not have committed any storage resources. In order to avoid the case in which the exponentials may be modified, the server uses a secret hash key HK to compute an authenticator $H_{HK}(g^y, (g^x)_P)$. The key HK is private to the server and is updated frequently, and the authenticator sent back in message (B.3) can be used to ensure that the exponentials are the same as those in (B.2). On the other hand, the server is also protected from computation DoS attacks aiming to exhaust the server's computation resource. In message (B.2), the server's computation cost includes only a cryptographic hash operation and an exponential g^y. Note that the random exponential g^y can be computed beforehand or periodically computed when the server is lightly computational burdened. Moreover, when the server is under the computation DoS attack and heavily burdened in computation, the server can reuse previously used exponential g^y. While if the adversary launches a computation DoS attack by flooding message (B.3) to the server, the server just resends message (B.4) to the other party.

Only the authorized client who has the valid password P can establish a secret key K with the authentication server and in turn with an access router. Since only the client with K can compute ICV for an IP packet, only the client can access the network services. Note that the ICV is computed over the MAC header and the entire IP packet; hence, any modification to the MAC header and the IP packet during transmission will be detected by the access router. This ensures that only authorized parties can gain access to the wireless network.

E. Performance Evaluation of Our Protocol

Compared with other protocols for access control in wireless networks, our protocol fulfills all the security requirements while not requiring certificates on the client end. A comparison between our protocol and other protocols is given in the table II. Only our protocol, the JFK protocol and the IKEv2 protocol provides both DoS resistance, but JFK and IKEv2 require certificates on the client end which is a big hurdle for implementation. Although EAP-TTLS, PEAP, LEAP, and EAP-PAX do not require client certificates, but they are vulnerable to DoS attacks. Moreover, LEAP and EAP-PAX are susceptible to dictionary attacks.

While providing desirable features for wireless networks, our protocol also achieves great computation efficiency for wireless networks. In our protocol, the server only needs 2 exponentiations and 1 public key decryption, and the client requires to compute 2 exponentiations and 1 public key encryption. We evaluate the performance of our protocol by measuring the overhead of our protocol. The overhead incurred by our access control protocol consists of two parts. The first part of the overhead comes from authentication and key exchange of the protocol. Before a client can access the wireless network, it needs to follow the protocol with the authentication server to agree on a session key for each session. This part of overhead is associated with every session. Thereafter, the client uses the session key to encrypt and authenticate every packet, while the access router verifies every packet from the client with the same session key. The delay of the packet processing leads to the second part of the overhead for the wireless network, and it is associated with every packet.

We adopt the benchmarks for the cryptographic operations on two different hardware platforms. One is a 450MHz Pentium III processor [23], [1], [21], and the other is a 2.1GHz Pentium IV processor [24]. We assume the following system setup for performance evaluation of our protocol: AES is used for encryption and HMAC/MD5 is used for ICV calculation; the bandwidth of the wireless network is 1Mb/s; the random nonces N_C and N_S, the exponents, the identity of each party are 160-bit long; the modulus p is 1,024-bit long.

For the first part of the overhead, the time on hash computation, random number generation, modular multiplication and modular inversion can be ignored, since it is relatively much smaller than the time on Diffie-Hellman key-pair generation and key agreement.

After successful authentication and key exchange, the client obtains the session key to secure its subsequent communications. With the session key, every packet is encrypted and an integrity check value (ICV) of the packet is calculated for the purpose of authentication. After the packet is received by the access router, the router decrypts the packet and computes the ICV of the packet for authentication. These operations incur the second part of overhead for our protocol.

We evaluate the performance of our protocol for packets of size 1000 bytes. The total overhead of our protocol is calculated and listed in table III. As seen from the table, the total overhead of our protocol takes up only 3.8% for the 450MHz Pentium III and 2.9% for the 2.1GHz Pentium IV of the total time. Therefore, our protocol can be used to secure wireless communications with degrading the performance slightly.

TABLE III

OVERHEAD OF OUR PROTOCOL FOR WIRELESS NETWORKS

	450MHz P III	2.1GHz P IV
Overhead/Session	53.0ms/8.4ms [1]	19.6ms/8.4ms
Overhead/Packet	0.08ms/0.16ms	0.04ms/0.16ms
Total Overhead [2]	$(61.4{+}0.24\cdot n)$ms	$(28.0{+}0.20\cdot n)$ms
Total Overhead [3]	3.8%	2.9%

[1] The data is in form of computation/transmission time.

[2] The total overhead for n-packet transmission.

[3] The total overhead of a 1000-packet session as a percentage of the total time.

IV. CONCLUSIONS

Security issues are crucial for wireless communications, and a secure and efficient access control mechanism is the first line of defense for secure wireless networking. In this paper we reviewed existing access control schemes for wireless networks, and identified their weaknesses and drawbacks. Then we proposed a new access control protocol that meets all the security requirements of wireless networks: mutual authentication, key confirmation, perfect forward secrecy, security against dictionary attacks, DoS resistance, and client identity confidentiality. We also present the security analysis and performance evaluation for our protocol, which show that our protocol is both secure and efficient for access control in wireless networks.

REFERENCES

[1] K. Aoki and H. Lipmaa, "Fast Implementations of AES Candidates," *Third AES Candidate Conference*, New York City, USA, 13–14 April 2000.

[2] W. A. Arbaugh, N. Shankar, and J. Wang, "Your 802.11 Networks Has No Clothes," in *Proceedings of the First IEEE International Conference on Wireless LANs and Home Networks*, December 2001.

[3] A. Aziz and W. Diffie, "Privacy and Authentication for Wireless Local Area Networks," *IEEE Personal Communications,* First Quarter:25-31, 1994.

[4] M.J. Beller, L.-F. Chang, and Y. Yacobi, "Privacy and Authentication on a Portable Communications System," *IEEE Journal on Selected Areas in Communications*, 11:821-829, 1993.

[5] S.M. Bellovin,"Problem Areas for the IP Security Protocols," in *Proceedings of the 6th USENIX Security Symposium*, San Jose, California, July 1996.

[6] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11," in *Proceedings of the th 7th Annual International Conference on Mobile Computing and Networking*, July 16-21, 2001.

TABLE II

A SUMMARY COMPARISON BETWEEN OUR PROTOCOL AND OTHER PROTOCOLS

Protocols	EAP-TLS	EAP-TTLS	PEAP	LEAP	EAP-PAX	JFK	IKEv2	Lancaster	Our Protocol
DoS Resistance	No	No	No	No	No	Yes	Yes	No	Yes
Anonymity	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Resistant to Dictionary attacks	Yes	Yes	Yes	No	No	Yes	Yes	No	Yes
Require Client Certificates	Yes	No	No	No	No	Yes	Yes	No	No

[7] S.M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure against Dictionary Attacks," in *Proceedings of the Symposium on Security and Privacy*, pages 72-84. IEEE, 1992.

[8] S.M. Bellovin and M. Merritt, "Augmented Encrypted Key Exchange: A Password-Based Protocol Secure against Dictionary Attacks and Password File Compromise," in *Proceedings of the 1st Annual Conference on Computer and Communications Security, ACM*, 1993.

[9] C. Boyd and D.-G. Park, "Public Key Protocols for Wireless Communications," in *Proceedings of the 1998 Internatinal Conference on Information Security and Cryptology (ICISC'98)*, 1998.

[10] A. Friday et al., "Network Layer Access Control for Context-Aware IPv6 Applications," *Wireless Networks*, vol. 9, pages 299-309, 2003.

[11] W. Aliello et al., "Just fast keying (JFK)," *IETF Draft(work in progress)*, draft-ietf-ipsec-jfk-04.txt, July 2002.

[12] D. Jablon, Strong Password-Only Authenticated Key Exchange, *ACM Computer Communications Review*, vol.26, no.5, 1996.

[13] C. Kaufman, "Internet Key Exchange (IKEv2) Protocol," *IETF Draft (work in progress)*, draft-ieft-ipsec-ikev2-14.txt, June 2004.

[14] S. Kent and R. Atkinson, "IP Authentication Header," *IETF Standards Track RFC 2402*, November 1998.

[15] T. Kwon, "Authentication and Key Agreement via Memorable Password," in *Proceedings of the ISOC NDSS Symposium*, 2001.

[16] A. Mishra and W. A. Arbaugh, "An Initial Security Analysis of the IEEE 802.1X Standard," Technical Report CS-TR-4328, UMIACS-TR-2002-10, University of Maryland, Febrary 2002.

[17] LAN MAN Standards Committee of the IEEE Computer Socciety, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification," *IEEE Standard 802.11,1997 Edition*, 1997.

[18] LAN MAN Standards Committee of the IEEE Computer Socciety, "Standards for Local and Metropolitan Area Networks: Standard for Port based network network access control," *IEEE Draft P802.1X/D11*, March 2001.

[19] LAN MAN Standards Committe of the IEEE Computer Society, "Amendment 6: Medium Access Control (MAC) Security Enhancements," *IEEE Standards P802.11i*, June 2004.

[20] P. MacKenzie, More Efficient Password-Authenticated Key Exchange, *Progress in Cryptology – CT-RSA 2001*, pages 361-377, 2001.

[21] B. Preneel et al., "Performance of Optimized Implementations of the NESSIE Primitives," *NESSIE Report*, Delivrable D12, February 2003.

[22] S. Schmid et al., "An Access Control Architecture for Microcellular Wireless IPv6 Networks," in *Proceedings of 26th IEEE Conference on Local Computer Networks (LCN'2001)*, pages 454-463, 2001.

[23] M. Scott, "Multiprecision Integer and Rational Arithmetic C/C++ Library ," Shamus Software Ltd, available at *http://indigo.ie/ mscott/*.

[24] W. Dai, "Crypto++ 5.1 Benchmarks," available at *http://www.eskimo.com/wei-dai/benchmarks.html*.

[25] T. Wu, "Secure Remote Password Protocol," *ISOC Network and Distributed System Security Symposium*, 1998.