Fast Authentication for Mobile Hosts in Wireless Mesh Networks

Celia Li and Uyen Trang Nguyen

Technical Report CSE-2010-03

March 29 2010

Department of Computer Science and Engineering
4700 Keele Street, Toronto, Ontario M3J 1P3 Canada

FAST AUTHENTICATION FOR MOBILE HOSTS IN WIRELESS MESH NETWORKS

Celia Li and Uyen Trang Nguyen

Department of Computer Science & Engineering
York University
4700 Keele Street, Toronto, Ontario M3J 1P3 Canada
Email: {cli, utn}@cse.yorku.ca

ABSTRACT

We aim at extending IEEE 802.11s standards to implement fast hand-off to support real-time applications such as VoIP and audio/video conferencing. We propose a novel trust model that represents the trust relationships among the entities of a WMN, and new authentication protocols based on that model. A client and a mesh access point (MAP) mutually authenticate each other using one-hop communications. No central authentication server is required. Fast authentication for roaming from one MAP to another is supported by using tickets. Our performance and security analysis show that our proposed authentication protocols are efficient and resilient to various kinds of attacks.

1. INTRODUCTION

Wireless Mesh Networks (WMNs) form a new class of networks that has emerged recently. The major components of a WMN [1] are shown in Fig. 1, which consist of

• mesh points (MP). The MPs form a wireless *mesh backbone* to provide multi-hop connectivity from one mesh client (STA) to another or to the Internet.

• mesh access points (MAP). A MAP is a mesh point that also works as an access point, i.e., connects mesh clients to the WMN.

• mesh point portal (MPP). A MPP is a mesh point that also works as a gateway connecting the WMN to the Internet.

• mesh clients (STA). Mesh clients can be static (e.g., desktops, database servers) or mobile hosts (e.g., cell phone, PDAs).

A WMN is dynamically self-organized and self-configured, with nodes in the network automatically establishing and maintaining mesh connectivity among themselves. This feature brings many benefits to WMNs such as low installation cost, large-scale deployment, reliability, and self-management.

Authentication is essential in any service-oriented communication networks to identify and reject any unauthorized network access. Design and implementation of authentication protocols, or any security protocols in general, in WMNs are challenging due to the following issues:

• Wireless channels have limited bandwidth and are error-prone.

• Wireless multi-hop routing drastically reduces network throughput [2].

• The shared broadcast medium makes the network vulnerable to several types of attacks such as eavesdropping, jamming, and packet interception and modification.

• Distributed network architectures make protocol design and implementation difficult.

• Mobile devices (e.g., cell phones, PDAs) have limited storage, computing capability and power supply.

• Clients' mobility requires efficient, fast hand-off mechanisms.

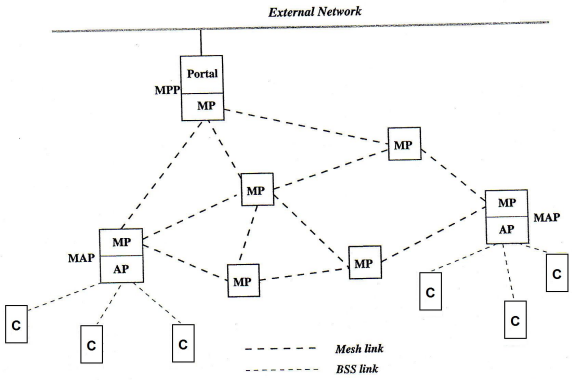

Fig. 1. Network architecture of an 802.11s WMN

Existing authentication protocols employed for wireless networks such as those in IEEE 802.11i and 802.11s standards do not meet the above needs and challenges. For instance, the authentication protocol in 802.11i is a centralized scheme (intended for use in wireless local area networks) and requires an authentication server. The protocol assumes one-hop wireless communications between mobile devices and an access point. Communications between the access points and the authentication server are multi-hop via *wired* links. In a WMN, this scheme is not suitable or efficient. First, multi-hop routing between an access point and the authentication server via *wireless* links would result in long delay, low reliability and thus potential service interruption. Second, a cen-

tral authentication server is not efficient in WMNs because the operations should be distributed for scalability.

IEEE 802.11s [3] defines standards for wireless mesh networks, and employs the same security architecture as 802.11i. That is, clients are also authenticated by an authentication server. The hand-off delay when clients move within the mesh can be large due to channel scan security, authentication, and other necessary operation procedures. For an 802.11 mesh, voice over IP (VoIP) is one of the killer applications. However, without a fast roaming scheme, it is impossible to deliver VoIP traffic without service disruption. Yet, the current version 802.11s does not specify any mechanisms/protocols that support fast hand-off.

Our work in this paper contributes towards extending the IEEE 802.11s standards to support fast roaming for mobile clients. In particular, we focus on fast authentication during the hand-off process as well as during initial login time.

We extend the capability of IEEE 802.11s by allowing mobile clients' authentications to be done by mesh access points (MAPs) and avoiding multi-hop communications with a central authentication server. Our proposed handover authentication protocol supports fast authentications from one MAP to another (i.e., during hand-off) to support client mobility in real-time applications (e.g., VoIP, stock quotes). In addition, our login authentication protocol improves the latency of authentication at login time compared with the 802.11i authentication protocol used by 802.11s. In particular, we propose

• a new trust model for WMNs based upon which our proposed authentication protocols are designed;

• ticket-based [4] authentication protocols that are efficient and resilient to attacks. No central authentication server is needed. Fast authentication during the hand-off process is supported using tickets [4].

The remainder of the paper is organized as follows. We discuss related work in Section 2. The proposed trust model and ticket design are described in Section 3. In Section 4, we present our login and handover authentication protocols, along with a security analysis of the protocols. A performance analysis of the proposed protocols in Section 5 shows that our login authentication protocol improves the latency of 802.11s login authentication, and our handover authentication protocol supports fast authentication during the hand-off process. Section 6 concludes the paper and outlines our future work.

2. RELATED WORK

In this section, we summarize existing work on trust management and authentication.

2.1. Trust Management

As an important concept in network security, trust is interpreted as a set of relations among entities participating in the network activities. A range of trust management schemes have emerged to satisfy the needs of the Internet. IBM research laboratory developed a trust establishment framework [5] allowing the bottom-up emergence of a public key infrastructure through exchange of certificates. Pretty Good Privacy (PGP) adopts the "web of trust" approach [6]. In this approach, there is no central authority that everybody trusts, but instead, individuals sign each other's public key certificates and progressively forming a web of individual public keys interconnected by links formed by their signatures. For example, Alice signs Bob's public-key certificate which she knows is authentic. Bob then forwards his signed certificate to Carol who wishes to communicate with Bob privately. Carol, who knows and trusts Alice as an introducer, finds out, after verification, that Alice is among Bob's certificate signer (Bob could have more than one signature on his certificate to make it more widely acceptable). Therefore, Carol can be confident that Bob's public key is authentic. The PGP "web of trust" is fully peer-to-peer and is efficiently used for the Internet.

Trust management in resource-constraint networks, such as mobile ad hoc network (MANETs) is much more difficult but more crucial than in the Internet [7]. This type of distributed networks have neither pre-established infrastructure, nor centralized control servers or trusted third parties. The trust information used to evaluate trustworthiness is provided by peers, i.e., the nodes that form the network. Resources, such as power, bandwidth and computation, are normally limited because of the wireless and ad hoc environment. Thus, the trust evaluation procedures should be efficient and only rely on local information. Each node, as an autonomous agent, makes the decision on trust evaluation individually. The decision is based on information it has obtained by itself or from its neighbors.

The architecture of a WMN is different from that of the Internet or MANETs. For example, it requires trust management among mesh access points (MAPs) of the mesh backbone; trust management between a client and a MAP to which the client is connected; and potentially trust management among clients for extended ad hoc networking. To manage the relationships among all these entities, a new trust model for WMNs is required upon which our proposed authentication protocols are designed.

2.2. Authentication

To analyze existing work on authentication in relation to WMNs, we first need to identify the requirements of an authentication protocol in WMNs.

• The protocol must incur *low* computation and communication costs. Mobile clients such as cell phones and PDAs typically have limited computational capabilities, storage and/or power supply. The computational loads imposed on these devices (e.g., encryption and decryption) should be kept as low as possible. The available bandwidth between a client and

its MAP can be limited, especially in 802.11-based networks; thus the number of messages to be exchanged should be minimized.
• The delay of re-authentication during the hand-off process should be low to avoid service interruption, especially in real-time applications such as VoIP.
• The operations must be distributed for scalability.
• The protocol must support mutual authentication (e.g., between a client and a MAP), protection of client identity privacy, and resilient to various types of attacks [8] such as source substitution attack, timememory trade-off attack, known key attack, etc. These types of attacks will be defined and discussed in Section 4.3.

Based on the above requirements, we now discuss existing work on authentication.

Several authentication protocols have been proposed for wired networks such as Kerberos [4] and SSL [9]. Kerberos uses symmetric key methods, which are ideal for network environments where all services and clients are known in advance. This is usually not the case in a WMN where clients may join, leave and move freely at will.

SSL uses public key methods, in particular public key certificates, to perform authentication, which is ideal for secure communications with a large, variable user base that is not known in advance, such as the Internet. However, public key methods are computationally intensive and space consuming, which are not suitable for resource-constrained mobile devices.

Standards for wireless networks includes IEEE 802.11i (WLANs) and 802.11s (WMNs). As mentioned earlier, 802.11i authentication is not efficient or scalable for use in WMNs due to its centralized operations. IEEE 802.11s inherits the security architecture from 802.11i, and thus suffers from the same drawbacks. In particular, there is no support for fast re-authentication, or fast hand-off in general, when a client moves from one MAP (or network) to another [3].

There exist also authentication protocols that support clients roaming from one network/domain to another in mobile IP and cellular networks [11, 10]. In this case, the foreign agent / network must communicate with a client's home agent/network in order to authenticate the client. As discussed before, wireless multi-hop routing in WMNs incurs long latency and potentially low reliability, which are not suitable for real-time applications such as VoIP and tele-conferencing.

The objective of our proposed authentication protocols is to support fast authentication during the login time as well as the hand-off process. The protocols require a new trust model and the use of tickets, which are described next.

3. PROPOSED TRUST MODEL AND TICKET TYPES

We present the definition of ticket and the trust model upon which our authentication protocols are built. We also describe in detail the different types of tickets used in the proposed authentication protocols.

3.1. Tickets

Our proposed trust model is based on the concept of ticket from Kerberos and a Kerberos-assisted authentication scheme proposed by Pizada and McDonald for mobile ad-hoc networks [12].

A ticket serves as a pass that a user submits to a system/network to allow it to verify the user's identity. One Kerberos ticket can be used for multiple services in the same system/network. Within the lifetime of a ticket, only a one-time authentication using password is required. As a result, tickets offers better security, more convenience and faster authentication then traditional authentication schemes using passwords [13].

Kerberos, however, is a centralized authentication scheme and not suitable for use in WMNs where distributed operations are desirable. For example, a Kerberos ticket is bound to the network that issues the ticket. A client must present its ticket to each network it visits. The home authentication server has to be involved for verifying the ticket and authenticating the client. In wireless multi-hop routing environments such as inter-domain mesh networks, the communication between the client in a foreign network and the home authentication server may incur unacceptable delay and service interruption while the client roams among networks. Our proposed trust model, ticket design and authentication protocols aim at minimizing the latency of the handover authentication process and service interruption.

3.2. Trust Model

The proposed trust model (shown in Fig. 2) is built upon the concept of "ticket" and "ticket agent". In this paper, a ticket agent is defined as an authority who issues and manages various types of tickets and can be trusted by various entities in a mesh network. There can be several ticket agents serving a network. Tickets are used to establish the trust relationships among entities.

Following are the trust relationships among the network entities shown in Fig. 2

1. Ticket agents: Different ticket agents establish mutual trust via their public key certificates issued by a Central Authority (CA).

2. MAPs: Any two neighboring MAPs trust each other via their shared symmetric key. This trust allows a client to roam among different MAPs in a mesh network.

3. MAPs and clients: The mutual trust relationship between a client and its home MAP is established via their respective client ticket and MAP ticket, which are described in Sections 3.3.1 and 3.3.2.

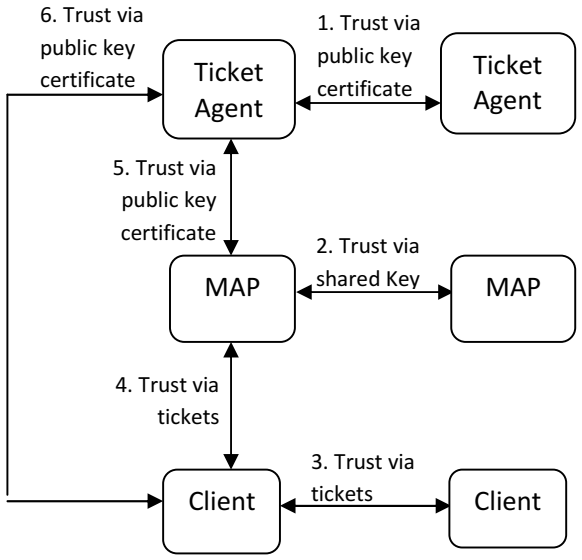

Fig. 2. Trust model of WMNs

4. MAPs and ticket agents: The mutual trust between a MAP and its ticket agent is established via their public key certificates issued by the CA. The trust is established when a MAP applies for a MAP ticket from a ticket agent.

5. Ticket agents and clients: The mutual trust is based on their public key certificates issued by the CA and is established when a client applies for a client ticket from a ticket agent.

3.3. Tickets in the Trust Model

Three types of tickets are used in our authentication protocols: client ticket, MAP ticket and transfer ticket. They are needed for mutual authentication between a client with a MAP when the client signs in the network or roams to another MAP.

We will use the notations listed in Table 1 throughout the paper to facilitate the discussions.

3.3.1. Client Tickets

A client applies for a client ticket from a ticket agent. The trust between a client and a ticket agent is established through their public key certificates issued by a CA.

Following is the structure of a client ticket:

$$\boxed{T_C = \{I_C, I_A, \tau_{exp}, P_C, Sig_A\}}$$

- T_C: client ticket issued by ticket agent I_A.
- I_C: ID number of the client that is given this ticket.
- I_A: ID number of the ticket agent who issued the ticket T_C.

Table 1. Notations

Notation	Description
C	Client
R	Mesh access point (MAP)
A	Ticket agent
I_x	ID number of entity x
Θ_C	Transfer ticket issued to a client
P_x	Public key issued to x
T_x	Ticket issued to x
τ_{exp}	Expiry date and time of a ticket
N_x	A nonce generated by x
Sig_x	Digital signature of entity x
MAC_{alg}	Type of MAC algorithm
$E_{pub_x}(m)$	Encryption of message m using x's public key
K_{MAC}	The key used to produce a message authentication code (Section 3.3.3)
$V_{K_{MAC}}(m)$	Message authentication code (MAC) resulting from the application of a MAC algorithm and a MAC key K_{MAC} on a message m

- τ_{exp}: expiry date and time of ticket T_C.
- P_C: public key of client I_C, which is used by a MAP to verify the signature signed by the client in the login authentication protocol (see Section 4.1).
- Sig_A: digital signature of ticket agent I_A, which gives a recipient reason to believe that the ticket was created by ticket agent I_A, and that it was not altered in anyway.

3.3.2. MAP Tickets

The operator of a mesh network applies for MAP tickets, one per MAP, and distributes them to the MAPs in the network. The operator is also responsible for requesting and distributing new MAP tickets before the current MAP tickets expire.

Following is the structure of a MAP ticket:

$$\boxed{T_R = \{I_R, I_A, \tau_{exp}, P_R, Sig_A\}}$$

- T_R: MAP ticket issued by ticket agent I_A.
- I_R: ID number of the MAP that is given this ticket.
- I_A: ID number of the ticket agent who issued ticket T_R to MAP I_R.
- τ_{exp}: expiry date and time of ticket T_R.
- P_R: public key of MAP I_R, which is used by clients to verify the signature of beacons message sent by MAP I_R (see Section 4.2).
- Sig_A: digital signature of ticket agent I_A.

3.3.3. Transfer Tickets

A transfer ticket is used to establish the trust relationship between a MAP and a client when a client roams from one MAP to another. When a client device C first logs in into the network, it submits its client ticket to a nearby MAP M_1, which will authenticate the client. If the authentication succeeds, M_1 becomes the *home MAP*[1] of C. At the end of the authentication, M_1 issues to C a transfer ticket and a secret key K_{MAC}. See step (1) in the diagram shown in Fig. 3, which shows the messages exchanged between the MAPs and client. When C roams to another MAP M_2, which we call *foreign MAP*, it submits the transfer ticket to M_2 for authentication. The transfer ticket proves to the foreign MAP that client C has been successfully authenticated by its home MAP.

The structure of a transfer ticket Θ_C is as follows:

$$\boxed{\begin{array}{l}\Theta_C = \{\mu, V_{K_{MAC}}\}, \text{where} \\ \mu = \{I_R, I_C, I_A, \tau_{exp}, MAC_{alg}\}\end{array}}$$

In the transfer ticket message μ stores the information of the client, home MAP and ticket agent as follows:

- I_R: ID number of the MAP who issues this transfer ticket.
- I_C: ID number of the client who owns this transfer ticket.
- I_A: *ID number of the ticket agent who issued C's client ticket.*
- τ_{exp}: expiry date and time of this ticket.

In addition, message μ contains the type[2] of Message Authentication Code (MAC) algorithm [14], which a foreign MAP will use in combination with value $V_{K_{MAC}}$ to verify the authenticity and integrity of the transfer ticket submitted by client C. The operation of and the need for the MAC algorithm are explained below.

When client C moves into contact with a foreign MAP M_2, to prepare for a hand-over to the new MAP, C submits the transfer ticket issued by M_1 to M_2 for authentication (step (3) in Fig. 3). This hand-over authentication requires the following additional cryptography operations and keys:

• A shared key[3] between M_1 and M_2, which allows M_1 to *securely* send a message r containing the ID of client C and the secret key K_{MAC} to M_2 (step (2) in Fig. 3). (This K_{MAC} key is the same key that the home MAP sent to C at the end of the login authentication process or step (1) in Fig. 3.)

• Before sending the transfer ticket to client C, the home MAP M_1 applies the MAC algorithm to message μ to produce a message authentication code denoted by $V_{K_{MAC}}$ (see Fig. 4). M_1 then combines message μ and $V_{K_{MAC}}$ to form the transfer ticket to be sent to C.

• Upon receiving both the message r sent by M_1 and the transfer ticket sent by C, M_2 verifies the authenticity and data integrity of the transfer ticket Θ_C by applying the MAC algorithm [14] to message μ in Θ_C using the key K_{MAC} to produce a MAC. If this MAC matches $V_{K_{MAC}}$ stored in the transfer ticket, then M_2 concludes that the ticket submitted by C is authentic. (In order for M_2 to further verify the identity of C, the hand-over authentication protocol requires additional steps, as will be discussed in Section 4.2. Those steps allow client C to use the key K_{MAC} it received from the home MAP during the log-in authentication.)

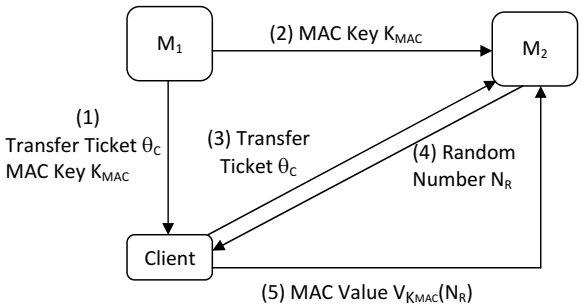

Fig. 3. Handover authentication

The log-in authentication protocol uses public-key cryptography and digital signatures, which are computationally intensive (but are done only once at the login time). To allow for fast authentication during handover, we use a MAC algorithm for transfer tickets instead of public-key cryptography and digital signatures. The use of a light-weight MAC algorithm for handover authentication is possible thanks to the design of transfer tickets and the trust relationships defined in Section 3.2 (in addition an existing key management scheme between neighboring MAPs [16]).

3.3.4. Ticket Design Analysis

We consider two critical factors in the design of tickets for authentication protocols in WMNs: low cost and security. The cost includes both computation and communication costs. Tickets must be able to resist possible attacks such as forgery and modification [17].

Low Cost

The login authentication protocol uses public-key methods, which are computationally intensive. However, the protocol is

[1] We borrow the terminology from mobile IP.

[2] The inclusion of the type of MAC algorithm in a transfer ticket is optional. It is not required if the parties agree on an algorithm in advance.

[3] Independently of authentication, a shared key is required between any two communicating MAPs in a mesh network, for encrypting/decrypting packets exchanged between them to combat attacks such as eavesdropping. This is called "key management" in wireless networks [15]. Our proposed authentication protocols simply use that shared key and the implemented key management protocol.

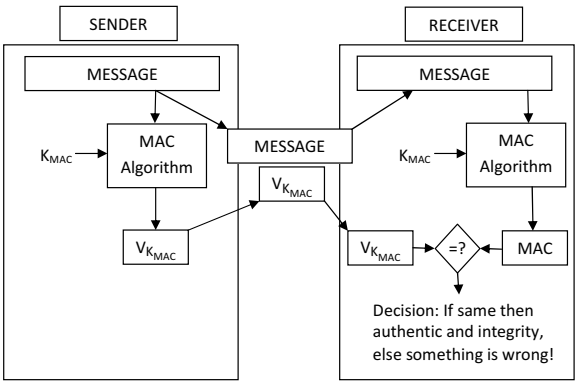

Fig. 4. Operations of a MAC algorithm

executed only once, when the client logs in into the network, in order to keep the computation load imposed on a client to a minimum.

A transfer ticket may be used several times by the client to roam from MAP to MAP in the network. Therefore, the computation load of handover authentication should be minimized. We use a MAC algorithm instead of public-key methods in the handover authentication protocol to achieve this goal.

Security

Possible attacks on tickets are forgery and modification. Forgery is the act of making or imitating tickets with the intent to deceive. Modification is the act of modifying the content of a ticket. We must prevent attackers from constructing or modify a ticket such that the network accepts it as a *valid* ticket.

Forgery: The proposed tickets are resilient to forgery attacks. A ticket agent signs a client (MAP) ticket using its private key. An attacker cannot generate a client (MAP) ticket without the knowledge of a ticket agent's private key.

A transfer ticket requires the application of a MAC algorithm and a MAC key to message μ to produce a MAC. Assuming a secure MAC algorithm such as HMAC [14], an attacker is not able to produce the correct MAC, or a valid transfer ticket, without the knowledge of the MAC key K_{MAC}. If we further assume that the MAC key K_{MAC} is securely transported from one MAP to another using their shared key and a strong encryption algorithm, then forgeries of transfer tickets are not feasible.

Modification: Similarly, the digital signature of a ticket agent ensures the authenticity and data integrity of client (MAP) tickets it generates. Assuming that the MAC algorithm is secure and MAC keys are safely transported, transfer tickets cannot be modified by an attacker and then accepted as valid.

(1) $R \longrightarrow C$: T_R
(2) $C \longrightarrow R$: $E_{pub_R}(M_C)$, where $M_C = \{T_C, N_{C_1}, N_{C_2}\}$
(3) $R \longrightarrow C$: $E_{pub_C}(M_R)$, where $M_R = \{N_{R_1}, N_{R_2}\}$
(4) $C \longrightarrow R$: $V_{K_{MAC}}(N_{R_2})$
(5) $R \longrightarrow C$: $\{V_{K_{MAC}}(N_{C_2}), \Theta_C\}$

Fig. 5. Login authentication protocol

4. PROPOSED AUTHENTICATION PROTOCOLS

We describe two proposed authentication protocols, one for the initial log in into a network and the other for subsequent roaming (hand-off/handover). We then analyze the security of the proposed authentication protocols.

4.1. Login Authentication

When a client C logs in a WMN, the client and its MAP R exchange their tickets and verify the validity of each other's ticket. The trust relationship between a client and a MAP is based on their exchanged tickets. (These two tickets can be issued by different ticket agents, as long as the client and the MAP trust each other's ticket agent.)

Following is the proposed login authentication protocol according to the order of the messages to be exchanged as shown in Fig. 5.

(1) A MAP R periodically broadcasts beacon messages which contains its MAP ticket. These beacon messages allow a client C to detect its presence in order to join the MAP. Client C verifies the digital signature of the ticket agent A who issued the MAP ticket T_R using A's public key. C also verifies other information in the MAP ticket such as the ID of the ticket agent and the ticket expiry date.

(2) If the above verifications are successful, C extracts the MAP's public key from the MAP ticket T_R (see Section 3.3.2) and generates a message M_C which contains C's client ticket T_C and two nonces N_{C_1} and N_{C_2}. C then encrypts the message using the MAP's public key and sends the encrypted message to the MAP R.

Upon receiving the message, R decrypts it using its private key, and verifies the digital signature of the ticket agent who issued the client ticket T_C (using the ticket agent's public key). R then verifies other information recorded in the client ticket T_C such as the ID of the ticket agent who issued T_C and the ticket expiry date.

(3) If the above verifications succeed, MAP R retrieves the client's public key from ticket T_C (see Section 3.3.1), and generates a message M_C containing two random numbers N_{R_1} and N_{R_2}. R then encrypts message M_C using the client's public key, and sends the encrypted

message to client C. C will decrypt the message using its private key to retrieve N_{R_1} and N_{R_2}.

Both the client and the MAP then calculate their shared MAC key K_{MAC} by applying a hash function H (e.g., SHA-1 [18], SHA-2 [19], MD5 [20]) to the message $\{N_{C_1}||N_{R_1}\}$, where the operator $||$ denotes a concatenation, and N_{C_1} and N_{R_1} are the random numbers generated in steps (2) and (3) above. That is, $K_{MAC} = H(N_{C_1}||N_{R_1})$.

(4) Client C then uses the key K_{MAC} and applies a (predetermined) MAC algorithm on N_{R_2} (created in step (3)) to produce a message authentication code $V_{K_{MAC}}(N_{R_2})$, which C then sends to the MAP. Upon receiving this message authentication code, the MAP performs the same computation as C just did to produce a message authentication code $V'_{K_{MAC}}(N_{R_2})$. If $V'_{K_{MAC}}(N_{R_2}) = V_{K_{MAC}}(N_{R_2})$, then the MAP has successfully authenticated the client C, because only C has the knowledge of the shared key K_{MAC} and N_{R_2}.

(5) To allow the client to authenticate the MAP, R applies the MAC algorithm and key K_{MAC} on the random number N_{C_2} (generated by C in step (2)) to produce a message authentication code $V_{K_{MAC}}(N_{C_2})$. The MAP also creates a transfer ticket Θ_C for C, and subsequently sends a message containing both the message authentication code and the transfer ticket to C.

When this message reaches the client, C carries out the same MAC computation as the MAP did to obtain a message authentication code $V'_{K_{MAC}}(N_{C_2})$. If $V'_{K_{MAC}}(N_{C_2}) = V_{K_{MAC}}(N_{C_2})$, client C has successfully authenticated the MAP. C will use the transfer ticket Θ_C to roam in the network.

The random numbers are needed to combat replay attacks, as will be discussed in Section 4.3.

4.2. Handover Authentication

When a client C wishes to move from one MAP to another, e.g., from M_1 to M_2, it first sends a request to M_1 informing it of the intention [21]. M_1 subsequently sends to M_2 a message $r = \{I_C, K_{MAC}\}$ which contains the ID of C, I_C, and key K_{MAC} for use with the MAC algorithm and C's transfer ticket (see Section 3.3.3). C waits for some amount of time then sends its transfer ticket to M_2 to prepare for switching to M_2[4].

Following is the handover authentication protocol according to the order of the messages exchanged; see also Fig. 6.

[4] As an alternative implementation, M_2 acknowledges the receipt of message r using a broadcast. Upon hearing the acknowledgment, C submits its transfer ticket to M_2.

(1) $C \longrightarrow M_2$: $\{\Theta_C, N_C\}$
(2) $M_2 \longrightarrow C$: $\{V_{K_{MAC}}(N_C), N_R\}$
(3) $C \longrightarrow M_2$: $V_{K_{MAC}}(N_R)$

Fig. 6. Handover authentication protocol

(1) Client C sends its transfer ticket Θ_C and a nonce N_C to the foreign MAP M_2. Recall from Section 3.3.3 that a transfer ticket consists of two parts: the relevant information stored in a message μ and a message authentication code $V_{K_{MAC}}(\mu)$, which is the result of applying a MAC algorithm and a MAC key to message μ. Also, M_2 receives from the home MAP M_1 a message r storing the client's ID and the MAC key K_{MAC} which M_1 used to generate the message authentication code $V_{K_{MAC}}(\mu)$. M_2 verifies the content of the transfer ticket, especially the ID of the client's ticket agent and the ticket expiry date. It then applies the MAC algorithm and the MAC key received from M_1 to message μ to output a message authentication code $V'_{K_{MAC}}(\mu)$. If $V'_{K_{MAC}(\mu)} = V_{K_{MAC}}(\mu)$, M_2 concludes that the transfer ticket is valid (i.e., C was successfully authenticated by its home MAP).

The above verification, however, does not prove C's identity. The following steps enable M_2 to verify C's identity.

(2) M_2 uses the MAC algorithm and key K_{MAC} on the nonce N_C to produce a message authentication code $V_{K_{MAC}}(N_C)$, which M_2 sends to client C along with a nonce N_R.

When C receives the message $\{V_{K_{MAC}}(N_C), N_R\}$ from M_2, it performs the same MAC computation as M_2 did to obtain $V'_{K_{MAC}}(N_C)$. If this value matches $V_{K_{MAC}}(N_C)$, the client has successfully authenticated the foreign MAP.

(3) Client C then executes the MAC algorithm using the MAC key K_{MAC} it computed in step (3) of the log-in authentication (Section 4.1), and the nonce N_R as input. The result is a message authentication code $V_{K_{MAC}}(N_R)$, which C will send to M_2.

Upon receiving $V_{K_{MAC}}(N_R)$, M_2 repeats the same MAC calculation on N_R. If it obtains the same message authentication code as $V_{K_{MAC}}(N_R)$, then this proves C's identity since C is the only client who has the knowledge of the MAC key K_{MAC}.

The nonces N_C and N_R, together with the MAC key K_{MAC} shared among the client, home MAP and foreign MAP, allow the foreign MAP and the client to verify each other's identity.

It should be noted that

- The handover authentication protocol does not use digital signatures or public key cryptography for fast han-

dover, but rather a MAC algorithm, to minimize authentication latency.

- If the foreign MAP M_2 receives the transfer ticket Θ_C before the message $r = \{I_C, K_{MAC}\}$ from the home agent (Section 3.3.3), M_2 will not be able to verify the validity of the transfer ticket because it does not have the MAC key K_{MAC} in order to apply the MAC algorithm to the ticket. In that case, M_2 sends back an error message to C and C who will initiate a log-in authentication instead of handover authentication. In this worst-case scenario, the handover authentication reverts back to the current practice in WMNs (i.e., repeating the login authentication with the foreign MAP). However, with careful design of message distribution (as future work) and low to moderate mobility speeds, we expect that this worst-case scenario does not happen often (i.e., M_2 should receive message r before the transfer ticket Θ_C), and the proposed handover authentication will be employed in most cases.

- After M_2 receives message $r = \{I_C, K_{MAC}\}$ from the home MAP, it also propagates this message to its neighbors to prepare for client C's future move to another MAP, say M_3. M_3 will use message r and the transfer ticket submitted by C to authenticate C as described above.

4.3. Security Analysis of the Authentication Protocols

In this section, we describe the countermeasures implemented in the proposed authentication protocols against the attacks listed in [8] that are relevant to our protocols.

Identity privacy attack. Most people would like to remain anonymous while roaming in WMNs for privacy reasons. In a client ticket, the client is identified by an ID number assigned by the ticket agent when he/she applies for the ticket. Only this ID number is used in all subsequent communications, and not the person's descriptive identity (e.g. user name, real name). The client's descriptive identity is known to and can be traced back from the assigned ID number only by the ticket agent.

Replay attack. The attacker records messages of a successful authentication and replays these messages in an attempt to be successfully authenticated and gain access to the network. We prevent this type of attack by using message encryption and nonces[5]. Consider an example in which an attacker attempts to impersonate a MAP by capturing and retransmitting a beacon message in step (1) of the login authentication protocol in Fig. 5. The attacker should not be able to modify the content of the original MAP ticket in the beacon message, thanks to the issuing ticket agent's digital signature in the original MAP ticket. A client C will respond to the attacker's beacon message with a message $M_C = \{T_C, N_{C_1}, N_{C_2}\}$ encrypted using the legitimate MAP's public key. The attacker will not be able to decrypt this message since it does not have the corresponding private key. Without the knowledge of the nonces N_{C_1} and N_{C_2}, the attacker will not pass client C's verification in step (5). We can show in a similar manner that the replay of any message in the login or handover authentication protocol will fail the authentication.

Source substitution attack: An attacker may be able to get a client's public key from the client ticket and manage to obtain a public key certificate under the attacker's name using the stolen public key. This attack can be prevented by ensuring that the certificate authority insists on the proof of knowledge of the corresponding private key before issuing a public key certificate.

Time-memory trade-off attack: A time-memory trade-off attack circumvents exhaustive search by pre-computing and pre-storing a large amount of data. With pre-computation done offline, the time taken in the online stage is shortened at the expense of more memory required. In the field of cryptography, a time-memory trade-off attack can be used to determine the data for which a hashed version is available. For a given hashed value of a password, the attacker can use partially pre-computed values in the hash space of a cryptographic hash function to guess the password. In our proposed authentication protocols, we use a hash-based MAC algorithm based on SHA-2[6], which is currently among the most secure hash functions, and is employed in several widely used security applications and protocols [18].

5. PERFORMANCE ANALYSIS

We compare our proposed login and handover authentication protocols with the EAP-TLS protocol (summarized in Fig. 7). We choose EAP-TLS for the comparison because it is the authentication protocol in IEEE 802.11i, and the current version of IEEE 802.11s standards for WMNs inherits the security features of IEEE 802.11i. The performance is measured in terms of

• computation costs, which are the latencies (in milliseconds) incurred by the security operations such as encryption, decryption and hashing [22];

• communication costs, which indicate the number of messages exchanged between a MAP and a client to complete an authentication session.

5.1. Computation Costs

The protocols to be compared perform a subset or all of the following security operations:

• Encryption using public key (E_{pub})

[5] A nonce is a random number that is used only once.

[6] Although no attacks have yet been reported on the SHA-2 variants, a new hash standard, SHA-3, is currently under development for stronger security [18].

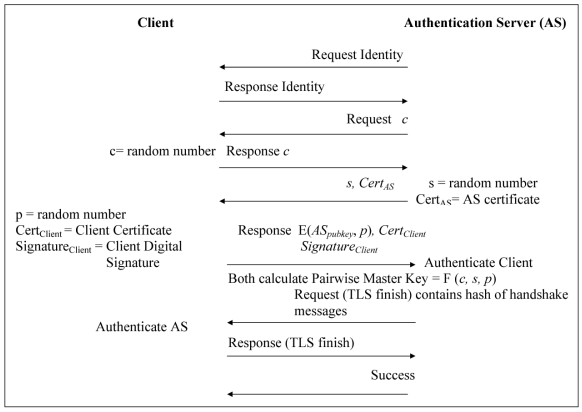

Fig. 7. EAP−TLS protocol

• Decryption using public key (D_{pub})
• Generation of a digital signature (G_{sig})
• Verification of a digital signature (V_{sig})
• Computation of a message authentication code (MAC)
• Hash

Table 2 lists the above operations, the current state-of-the-art algorithms implementing the operations, and the computation time each of these algorithms incurs [22] (the first, second and third columns, respectively).

The proposed login authentication protocol (Fig. 5) performs one public-key encryption, one public-key decryption, one signature generation, three signature verifications, one MAC operation and no hash function. The fourth column of Table 2 records the above numbers of operations. By multiplying the computation cost of each operation (from the third column) and the number of times it is executed, and summing up the costs of all operations the login authentication protocol performs, we obtain a total computation cost of 97.93 ms, as shown in the last row of the fourth column.

Similarly, the fifth and sixth columns of Table 2 list the numbers of security operations the proposed handover authentication protocol (Fig. 6) and EAP-TLS (Fig. 7) perform, respectively. Applying similar calculations as above, we obtain the computation costs of the proposed handover authentication protocol and EAP-TLS, which are 0.009 ms and 97.96 ms, respectively.

We can see that the computation cost of the login authentication protocol is slightly less than that of EAP-TLS. But more importantly, the computation latency of the handover authentication protocol is four orders of magnitude lower than that of the login authentication and EAP-TLS.

5.2. Communication Costs

Table 3 lists the number of messages involved in each of the three protocols we compare. The proposed login and handover authentication protocols require less messages to be exchanged than EAP-TLS, assuming one-hop communications between a client and a MAP.

5.3. Authentication Latency

The authentication latency T is defined as $T = T_c + dhT_m$, where

- T_c is the computation cost (in ms) of the protocol as given in Table 2 ;
- T_m is the communication cost in terms of the number of messages exchanged, as shown in Table 3;
- d is the average delay for one message/packet to be transmitted by one node and then received by a neighboring node (i.e., the average delay of a one-hop communication);
- h is the number of hops between the client and the authentication server. In our proposed authentication protocols, no authentication server is involved; all authentications are between the client and a nearby MAP; hence $h = 1$ (i.e., one-hop communications). In EAP-TLS, all authentications have to be performed by the home authentication server, requiring multi-hop communications in most cases, resulting in $h \geq 1$.

The authentication latencies of the three protocols are given in Table 4, and plotted in the graph in Fig. 8. The graph shows that the larger the number of hops between a client's home MAP and a foreign MAP, the lower the authentication latency our protocols incur compared with EAP-TLS. This contributes towards a faster hand-off process for real-time services.

Table 2. Computation costs

Op.	Alg.	Time (ms)	Login (Fig.5)	Handover (Fig.6)	EAP-TLS (Fig.7)
E_{pub}	RSA [23]	1.42	1	0	1
D_{pub}	RSA	33.3	1	0	1
G_{sig}	ECDSA [24]	11.6	1	0	1
V_{sig}	ECDSA	17.2	3	0	3
MAC	HMAC [14]	0.015	1	6	1
Hash	SHA-1 [18]	0.009	1	0	3
Total computational cost			97.93ms	0.009ms	97.96ms

Table 3. Communication costs

Protocol	Number of messages
Login	5
Handover	3
EAP-TLS	9

Table 4. Authentication latency

Protocol	Authentication latency
Login	$97.93 + 5d$
Handover	$0.009 + 3d$
EAP-TLS	$97.96 + 9dh$

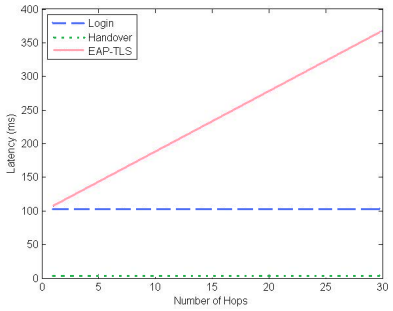

Fig. 8. Authentication latency comparison

6. CONCLUSION

The objective of our work is to extend the capabilities of IEEE 802.11s standards to support fast hand-off for real-time applications such as VoIP, tele-conferencing, and stock quote distribution. We propose a novel trust model that represents the trust relationships among the entities of a WMN, and authentication protocols based on that model. A client and a MAP mutually authenticate each other using one-hop communications. No central authentication server is required. Fast authentication for roaming from one MAP to another is supported by using tickets. The performance and security analysis show that our proposed authentication protocols are efficient and resilient to various kinds of attacks. In the future, we will carry out a comprehensive performance evaluation of login and handover authentication protocols in a WMN setting in comparison with the EAP-TLS authentication protocol of 802.11s using actual network performance metrics such as throughput, loss rate, end-to-end delay and delay jitter.

7. REFERENCES

[1] I. Akyildiz and X. Wang, Wireless Mesh Networks, Wiley, 2009.

[2] D. D. Couto, D. Aguayo, J. Bicket and R. Morris, "A High-Throughput Path Metric for Multi-hop Wireless Routing," ACM MobiCom, 2003.

[3] IEEE, "Draft Amendment: ESS Mesh Networking," IEEE 802.11s Draft 1.00, 2006.

[4] J. Kohl and C. Neuman, "The Kerberos Network Authentication Service (V5)," RFC 1510, 1993.

[5] A. Herzberg, "Access Control Meet Public Key Infrastructure, Or: Assigning Roles to Strangers," IEEE Symposium on Security and Privacy, 2000.

[6] P. R. Zimmermann, "The official PGP User's Guide," MIT Press, 1995.

[7] S. Yi and R. Kravets, "Key Management for Heterogeneous Ad Hoc Wireless Networks," IEEE International Conference on Network Protocols, 2002.

[8] G. Horn, M. Martin and C. Mitchell, "Authentication Protocols for Mobile Network Environment Value-Added Services," IEEE Transactions on Vehicular Technology, Vol. 51, No. 2, pp. 383-392, 2002.

[9] D.Wagner and B.Schneier, Analysis of the SSL 3.0 Protocol, The Second USENIX Workshop on Electronic Commerce Proceedings, pp. 29-40, 1996.

[10] Y. Jiang, C. Lin, X. Shen and M. Shi, "Mutual Authentication and Key Exchange Protocols for Roaming Services in Wireless Mobile Networks," IEEE Transactions on Wireless Communications, Vol. 5, No. 9, pp. 2569 - 2577, 2006.

[11] M. Buddhikot, G. Chandranmenon, S. Han, Y. Lee, S. Miller and L. Salgarelli, "Design and Implementation of a WLAN/CDMA 2000 Interworking Architecture," IEEE Communications Magzine, 2003.

[12] A. A. Pizada and C. McDonald, "Kerberos Assisted Authentication in Mobile Ad-hoc networks," CRPIT'04, The 27th Conference on Australasian Computer Science, Vol. 56, No. 41-46, Australia Computer Society, 2004.

[13] D. P. Jablon, "Password Authentication Using Multiple Servers," Topics in Cryptology - CT-RSA 2001, pp. 344-360, 2001.

[14] H. Krawczyk, M. Bellare and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104, 1997.

[15] A. J. Menezes, P. C. Oorschot and S. A. Vanstone, "Handbook of Applied Cryptography," CRC Press.

[16] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz and A. Khalili, "A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks," ACM Transactions on Information and System Security, Vol. 8, No. 2, pp. 228-258, 2005.

[17] J. Zhou, "A Generic Protocol for Controlling Access to Mobile Services", International Workshop on Applied Public Key Infrastructure (IWAP 2005), 2005.

[18] C. S. Jutla and A. C. Patthak, "Is SHA-1 Conceptually Sound?" Cryptology ePrint Archive, Report 2005/350, http://eprint.iacr.org/, 2005.

[19] P. Hawkes, M. Paddon and G. Rose, "On Corrective Patterns for the SHA-2 Family," Cryptology ePrint Archive, Report 2004/207, 2004.

[20] B. D. Boer and A. Bosselaers, "Collisions for the Compression Function of MD5," EUROCRYPT 1993, pp. 293-304, 1993.

[21] P. Goransson and R. Greenlaw, “Secure Roaming in 802.11 Networks,” Elsevier, 2007.

[22] M. Long, “Energy-efficient and Intrusion Resilient Authentication for Ubiquitous Access to Factory Floor Information,” IEEE Transaction on Industrial Informatics, Vol. 2, No. 1, pp. 40-47, 2006.

[23] R. Rivest, A. Shamir and L. Adleman, “A Method for Obtaining Digital Signatures and Public Key Cryptosystems,” Communication of the ACM, vol. 21, no. 2, pp. 120-126, 1978.

[24] ECDSA, FIPS 186-3, Digital Signature Standard (DSS), 2009.