PLOS ONE

RESEARCH ARTICLE

A New Ticket-Based Authentication Mechanism for Fast Handover in Mesh Network

Yan-Ming Lai[2], Pu-Jen Cheng[2], Cheng-Chi Lee[1,3]*, Chia-Yi Ku[2]

1 Department of Library and Information Science, Fu Jen Catholic University, New Taipei, Taiwan, 24205, R.O.C., **2** Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 10617, R.O.C., **3** Department of Photonics and Communication Engineering, Asia University, Taichung, Taiwan 413, R.O.C.

* cclee@mail.fju.edu.tw

Citation: Lai Y-M, Cheng P-J, Lee C-C, Ku C-Y (2016) A New Ticket-Based Authentication Mechanism for Fast Handover in Mesh Network. PLoS ONE 11(5): e0155064. doi:10.1371/journal.pone.0155064

Editor: Muhammad Khurram Khan, King Saud University, SAUDI ARABIA

Received: January 20, 2016

Accepted: April 23, 2016

Published: May 12, 2016

Data Availability Statement: All relevant data are within the paper.

Funding: This research was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract no.: MOST 104-2221-E-030-002. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. There was no specific funding for the rest of this project. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Due to the ever-growing popularity mobile devices of various kinds have received worldwide, the demands on large-scale wireless network infrastructure development and enhancement have been rapidly swelling in recent years. A mobile device holder can get online at a wireless network access point, which covers a limited area. When the client leaves the access point, there will be a temporary disconnection until he/she enters the coverage of another access point. Even when the coverages of two neighboring access points overlap, there is still work to do to make the wireless connection smoothly continue. The action of one wireless network access point passing a client to another access point is referred to as the handover. During handover, for security concerns, the client and the new access point should perform mutual authentication before any Internet access service is practically gained/provided. If the handover protocol is inefficient, in some cases discontinued Internet service will happen. In 2013, Li et al. proposed a fast handover authentication mechanism for wireless mesh network (WMN) based on tickets. Unfortunately, Li et al.'s work came with some weaknesses. For one thing, some sensitive information such as the time and date of expiration is sent in plaintext, which increases security risks. For another, Li et al.'s protocol includes the use of high-quality tamper-proof devices (TPDs), and this unreasonably high equipment requirement limits its applicability. In this paper, we shall propose a new efficient handover authentication mechanism. The new mechanism offers a higher level of security on a more scalable ground with the client's privacy better preserved. The results of our performance analysis suggest that our new mechanism is superior to some similar mechanisms in terms of authentication delay.

1 Introduction

With mobile devices coming to play a bigger and bigger part in our everyday lives, the need for wireless network systems to remain state of the art has become an indispensable urge if the service providers are to stay competitive. The wireless mesh network (WMN) is one of the best-

Competing Interests: The authors have declared that no competing interests exist.

known communication network architectures. It consists of mesh clients and mesh points. Mesh clients can be static hosts (e.g., desktops, servers) or mobile hosts (e.g., smart phones, laptops, and tablets), and they can access the Internet through mesh points. Due to its low cost, large-scale coverage, and high reliability, WMN is widely used nowadays. Several working groups (e.g., IETF) focus their attention on the development of WMN technologies, and corresponding specifications are being standardized (e.g., IEEE 802.12, 802.15 and 802.16).

Before accessing the Internet, a client must be authenticated by a mesh access point (MAP). When roaming from a mesh access point to another [1], as illustrated in Fig 1, the client needs to be re-authenticated to receive further Internet services. To keep real-time applications going and thus to offer the best user experience, the overall handover latency should not exceed 50*ms* [2]. However, the current wireless mesh networking standard IEEE 802.16m needs about 1000*ms* to process a full Extensible Authentication Protocol (EAP) for the overlong round trip between the client and the EAP server [3]. To make things worse, this same procedure has to be performed each time when a client moves to a new MAP (e.g. from MAP_1 to MAP_2) although the current EAP authentication has not yet expired. Obviously, there is plenty of room for improvement.

In order to reduce the latency during client roaming, quite a number of handover authentication protocols have been developed [4–21]. Among them, the earlier works focused on accelerating the full authentication mechanism with the authentication process still having to be repeated every time. Later on, some protocol developers decided that, after the first thorough authentication procedure, neighboring MAPs should pre-recognize the client, and thus the same client can later have a rapid pass by presenting a ticket when entering the realm of a new MAP. These ticket-based protocols mainly fall into three categories, which are handover single authentication, group key authentication, and broadcast authentication. Details of different types of ticket-based protocols will be elaborated in Section 2.

Recently, Li et al. proposed a fast handover authentication mechanism based on tickets for mesh network [22]. In spite of the efficiency and convenience it brings, Li et al.'s mechanism still has some weaknesses. In this paper, we shall present an efficient and secure authentication

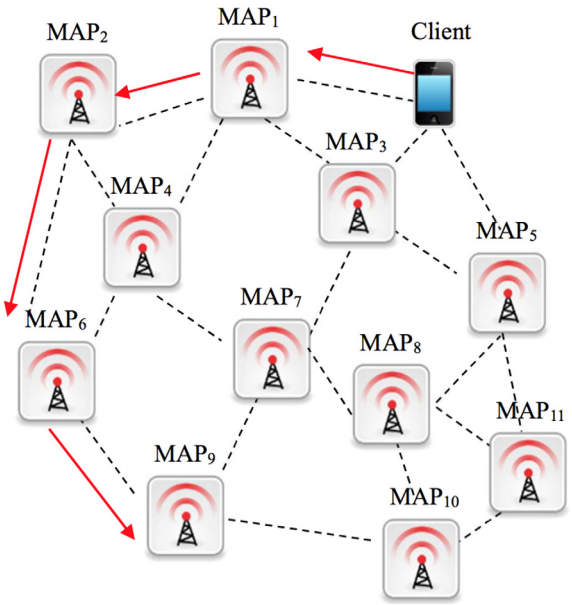

Fig 1. Wireless Mesh Network.

doi:10.1371/journal.pone.0155064.g001

mechanism we have developed to solve the problems that trouble Li et al.'s mechanism. The main contributions of this paper are as follows.

1. Distinct from the existing handover authentication schemes, the proposed new scheme saves the trouble of digital signature computations on the client side, which significantly reduces the computation cost and handover latency, making the scheme especially suitable for applications where the clients have mobile devices with limited computation power.
2. Unlike most existing handover authentication schemes, the proposed new scheme is applicable when the MAPs are not equipped with high-quality TPDs. Besides, our new scheme can keep another MAP from using the broadcast ticket to pretend to be the current MAP, and therefore there is absolutely no room for the domino effect to happen. In other words, our new scheme is not only more scalable but more secure.
3. In the proposed new scheme, privacy is well preserved, and the authentication imposes negligible overhead.

In this paper, we review Li et al.'s scheme and propose an improvement to refine the approach and make it become applicable. The rest of this paper is organized as follows. Section 2 reviews various ticket-based handover protocols and briefly introduces the basic underlying concepts. Section 3 gives some preliminaries that are to be used throughout this paper. Then, in Section 4, we review and analyze the Li et al.'s scheme. After that, we present an improved scheme in Section 5, followed by a security analysis and a performance analysis of the proposed scheme in Section 6 and Section 7, respectively. Finally, the conclusion will be in Section 8.

2 Related Work

Instead of accelerating the full authentication mechanism while repeating the authentication procedure every time, some protocols based on the Kerberos-style ticket [15] have been proposed. The main idea behind these methods is to reduce the authentication latency during the handover process by using a symmetric encrypted ticket. Only certified MAPs own the legal symmetric key and can generate a legal ticket for a verified client. For this reason, any client can submit a legal ticket to prove he/she has passed the authentication procedure with another certified MAP. This way, a legal MAP can simply handover authenticate a verified client. That means a client can wander all over the place receiving non-stop Internet service as long as the ticket has not expired yet. Based on the different mutual authentication mechanisms between client and MAP, we can mainly divide ticket-based authentication schemes into three types: single authentication, group key authentication, and broadcast authentication.

2.1 Single Authentication

In ticket-based handover by single authentication [13, 18, 19], it is assumed that each MAP has the pre-stored symmetric key shared among neighboring MAPs. Before a client steps off the coverage of some specific MAP (e.g. MAP_1), the client submits a handover single to MAP_1 to inform it as to which MAP (e.g. MAP_2) he/she will move to. Upon receiving the signal, MAP_1 uses $Key_{MAP1\text{-}MAP2}$ to generate a ticket and send this ticket to MAP_2. When the client arrives at MAP_2, MAP_2 can verify the client simply by using the ticket generated by MAP_1. However, it will get very confusing when the MAPs within the network are situated in a complicated way [23], for in that case it might not be very clear which MAP is the so-called MAP_2 that the client is moving on to.

2.2 Group Key Authentication

In ticket-based handover by group key authentication [6–10, 15, 18, 21], the authentication, authorization, and accounting (AAA) server sets up a multi-MAP group, and pre-distributes a private Multi-MAP Group Key (MGK) to each MAP in the group. Thus, a client does not need to inform MAP_1 which MAP he/she will move to. MAP_1 can use the general MGK to encrypt a ticket and then send it to the client. When the client arrives at MAP_2, he/she can readily submit the ticket to MAP_2 and get the service. This design based on a single group key, however, is neither secure nor scalable in large-scale mesh networks. To ensure the security of the single group key, each MAP in the group should be a high-quality TPD [24], assuming that they are secure against any compromise attempt in any circumstances. Unfortunately, this is too high a ground to reach [14]. In addition, ticket-based handover by group key authentication will not be an option when any MAP in the group might be malicious.

2.3 Broadcast Authentication

In ticket-based handover by broadcast authentication [4, 6, 14, 16, 17, 19], the AAA server maintains every MAP and its neighbors' locations. When a client is about to leave a MAP, the AAA server will send tickets to all the neighboring MAPs over a secure channel. However, this means there will be a very long latency time because the AAA server is normally many hops away from the client [3, 19]. To solve this problem, Li et al. proposed a fast handover authentication mechanism based on ticket broadcast for mesh network [22]. In their scheme, the task of pre-sending tickets is forwarded from the AAA server to the current MAP (e.g. MAP_1). In other words, when the client is about to leave, MAP_1 can use a pre-established symmetric key to generate tickets and send corresponding tickets to its neighbors (e.g. MAP_2, MAP_3, and MAP_4). This way, the ticket pre-distribution is completed right between the current MAP and its neighbors, only one hop across. However, in Li et al.'s scheme there are some leaks such as sending expiration time and date in plaintext and pre-sending the same ticket to all the neighboring MAPs. Later in Section 4 we shall give a thorough review of Li et al.'s scheme and detail the leaks we have found.

3 Preliminaries

In this section, we present some models and tools commonly used in this field. They include the trust model, different types of tickets, and elliptic curve cryptography (ECC).

3.1 Trust Model

The trust model is illustrated in Fig 2. A ticket agent (TA) is a trusted third party who generates and manages various types of tickets in a mesh network. The following are the elements shown in Fig 2:

- TA—mesh access points (MAPs): The mutual trust is based on public key cryptography and is built when a MAP requests a MAP ticket from TA. In response, TA embeds a digital signature in the MAP ticket to make MAPs believe the ticket was created by TA.
- TA—client: The mutual trust is based on public key cryptography and is built when a client requests a client ticket from TA. In response, TA embeds a digital signature in the client ticket to make the client believe the ticket was created by TA.
- MAP—client: The mutual trust relationship between a client and its home MAP is built through their respective client ticket and MAP ticket, which will be elaborated later.

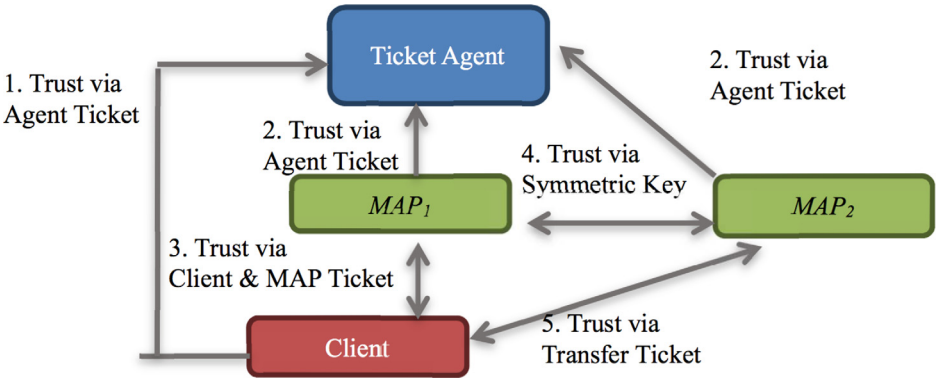

Fig 2. Trust model.

doi:10.1371/journal.pone.0155064.g002

- MAP_1—MAP_2: Any two neighboring MAPs build up mutual trust via symmetric key certificates. This trust allows a client to roam among different MAPs in a mesh network.

Please note that technically both the MAP ticket and the client ticket can be obtained before a client even joins a mesh network, so this portion of time consumption in public key operations should not count as part of the authentication process, which adds to the efficiency of the authentication protocol.

3.2 Different Types of Tickets

Three types of tickets are used in this paper: client ticket, MAP ticket, and transfer ticket. They are needed for mutual authentication between a client and a MAP when the client logs into the network or roams from a MAP to another. Before looking any further into the details of the three types of tickets, let's check out the notations listed in Table 1 below.

3.2.1 Client tickets. The client ticket (T_C) enables a client to gain a MAP's trust. The MAP (e.g. MAP_1) can verify the client by checking the client ticket to see if the ticket was really issued by TA.

Table 1. Notation table.

Notation	Description
C	Client
M	Mesh access point (MAP)
A	Ticket agent
P_x	Public key assigned to entity x
N_x	A nonce generated by entity x
Sig_x	Digital signature of entity x
$E_{Px}(m)$	Encryption of the message m by entity x's public key
$D_{Px}(m)$	Decryption of the message m by entity x's public key
$E_K(m)$	Encryption of the message m by a shared key K
$D_K(m)$	Decryption of the message m by a shared key K
K_{MAC}	The key used to produce a message authentication code
$V_{KMAC}(m)$	Message authentication code (MAC) of message m in combination with a secret shared key K_{MAC} [25]
$f(m)$	pseudo-random number generation function applied to message m
$H(m)$	Collision-free one-way hash function applied to message m [26]
||	A concatenation operation

doi:10.1371/journal.pone.0155064.t001

A client ticket T_C typically includes the following elements:

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

- I_C: ID of the client who keeps this ticket.
- I_A: ID of TA who generated this ticket.
- τ_{exp_C}: expiry time of T_C. When this ticket expires, the client needs to re-request a new ticket from TA.

3.2.2 MAP tickets. The MAP ticket (T_M) enables a MAP to gain a client's trust. The client (e.g. C) can verify the MAP (e.g. MAP_1) by checking the MAP ticket to see if the ticket was truly issued by TA.

A typical MAP ticket T_M includes the following elements:

$$T_M = \{I_M, I_A, \tau_{exp\_M}, P_M, Sig_A\}$$

- I_M: ID of the MAP who keeps this ticket.
- I_A: ID of TA who generated this ticket.
- τ_{exp_M}: expiry time of T_M. When this ticket expires, the MAP needs to re-request a new ticket from TA.

3.2.3 Transfer tickets. When a client C starts to access the mesh network, he/she needs to exchange his/her client ticket for the nearest MAP's (e.g. MAP M_1) MAP ticket to perform mutual login authentication. If the authentication phase is a success, M_1 issues a transfer ticket to C and becomes the home MAP of C. The transfer ticket helps to construct trust relationship between a foreign MAP (e.g. MAP M_2) and C. When C roams to M_2, he/she submits the transfer ticket to M_2 for handover authentication. With the transfer ticket, the client C can prove to M_2 that he/she has been authenticated by M_1. Thus, M_2 can run a simple authentication process to verify C.

The elements of a typical transfer ticket θ_C include:

$$\theta_C = \{I_C, I_M, I_A, \tau_{exp\_\theta}, V_{KMAC}(I_C, I_M, I_A, \tau_{exp\_\theta})\}$$

- I_C: ID of the client who obtained this ticket.
- I_M: ID of the home MAP who generated this ticket.
- I_A: ID of TA who issued I_C's client ticket.
- τ_{exp_θ}: expiry time of θ_C which is determined by its issuer's policy. When this ticket expires, the client needs to re-select a MAP to be the home MAP and re-login to obtain a new transfer ticket. Before that, the client can roam past MAP after MAP with ease.

3.3 Elliptic Curve Cryptography (ECC)

Elliptic curve cryptography (ECC) was created by Neal Koblitz and Victor Miller [27] in 1985, and it has been widely used nowadays [20, 28, 29]. Suppose G is a group of p members, where p is a large prime number. Let $\{a, b\} \in Z^*_n$ be such that $4a^3 + 27b^2 \neq 0 \pmod{p}$ in G. The set $E(G)$ consists of all points $(x, y) \in G$ that satisfy the equation $y^2 = x^3 + ax + b$, together with a special point O, namely the point at infinity. Then, let's select a q-order subgroup G_q of the additive group of points over $E(G)$ and choose an arbitrary generator P of G_q. According to the Elliptic curve discrete logarithm problem (ECDLP), given $mP \in E(G)$ and $P \in G_q$ for an unknown $m \in Z^*_n$, it is intractable to find m. Finally, we preload each client and MAP with the public system parameters $\{p, q, E(G), Z^*_n, P\}$.

4 Li et al.'s Authentication Protocols

In this section, we shall review Li et al.'s authentication protocols and present our analysis of their protocols. In Li et al.'s scheme [22], there are two distinct authentication protocols: initial login authentication protocol (LAP) and handover authentication protocol (HAP), as shown in Fig 3. These authentication protocols follow a key hierarchical structure similar to that in IEEE 802.11i. A pairwise master key (PMK) is created during the authentication process, and then a pairwise transient key (PTK) and a group transient key (GTK) are derived from the PMK. Because a typical mobile device has limited computation power, the number of message exchanges and that of public key operations should be kept down.

4.1 The Login Authentication Protocol (LAP)

Assume that the client and the MAP have respectively obtained a client ticket and a MAP ticket from TA. Now the pair of client and MAP submit tickets to each other for mutual authentication. The steps are as follows:

1. When a client C starts to access the mesh network, he/she broadcasts a request message containing his/her ID number to the neighboring MAPs.
2. Assuming MAP M_1 receives the request message, M_1 replies with a message containing its MAP ticket T_{M1} to inform C of its presence. After receiving T_{M1}, C verifies Sig_A. If the verification fails, C ignores this ticket. Otherwise, C checks the τ_{exp_M} of T_{M1} and determines whether or not the τ_{exp_M} has expired.
3. If the above verifications are successful, C extracts M_1's public key P_{M1} from T_{M1}. Then C encrypts ticket T_C along with two nonces N_{C1} and N_{C2} by using P_{M1}, and sends the encrypted message to M_1. Upon receiving the encrypted message, M_1 uses its own private key to decrypt the message and then verifies Sig_A in T_C. If the verification fails, M_1 ignores this ticket. After that, M_1 checks the τ_{exp_C} of T_C and determines whether or not it has expired.
4. If the above verifications are successful, M_1 extracts C's public key P_C from T_C. Then uses P_C to encrypt two nonces N_{M11} and N_{M12}. After that, M_1 sends the encrypted message to C and calculates their shared MAC key $K_{MAC} = N_{C1}||N_{M11}$ and pairwise master key $PMK_0 = N_{C2}||N_{M11}$. Upon receiving the message, C decrypts it by using his/her private key to obtain N_{M11} and N_{M12}. Then, C calculates their shared MAC key $K_{MAC} = N_{C1}||N_{M11}$ and pairwise master key $PMK_0 = N_{C2}||N_{M11}$. Due to the public-key cryptography, the security of nonces N_{C1}, N_{C2}, N_{M11}, and PMK_0 is ensured.

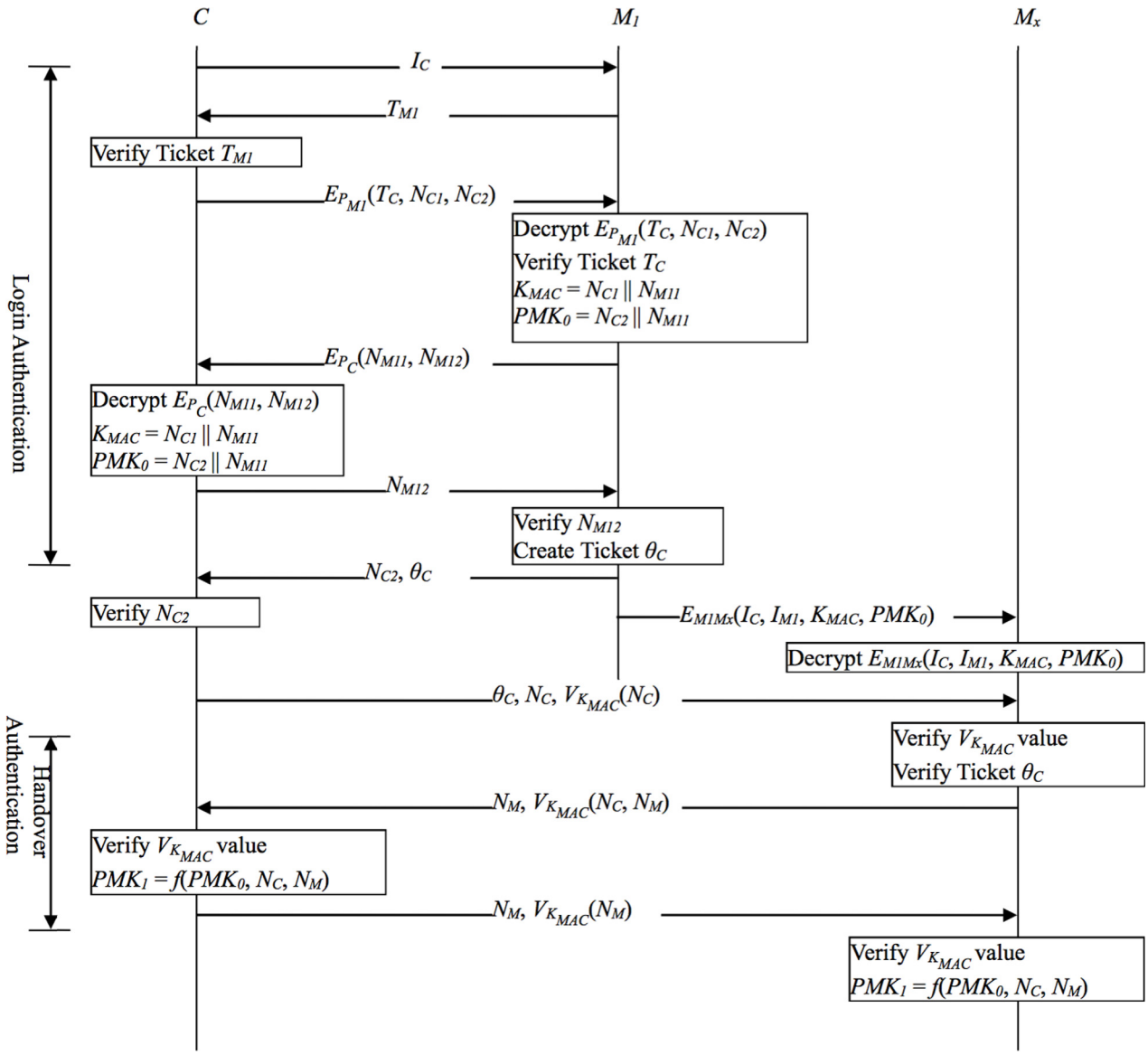

Fig 3. Li et al.'s authentication protocols.

doi:10.1371/journal.pone.0155064.g003

5. After that, C sends N_{M12} to M_1. Upon receiving this message, M_1 verifies N_{M12} by checking if it matches the value provided by M_1 itself earlier. If N_{M12} does not check out, M_1 ignores this message.

6. To complete the login authentication protocol, M_1 generates a transfer ticket θ_C and sends N_{C2} and θ_C to C. Upon receiving this message, C verifies N_{C2} by checking if it matches the value provided by C himself/herself earlier. If the N_{C2} value does not check out, C ignores this message.

This concludes the login authentication protocol. The pairwise master key PMK_0 can then be used to encrypt messages between the two parties. In the meantime, C can use this transfer ticket θ_C to legally roam from MAP_1 to another MAP in the network.

4.2 The Handover Authentication Protocol (HAP)

To support fast handover for clients roaming from M_1 to another MAP, M_1 should pre-distribute the keys shared with C to all the neighboring MAPs by broadcast. Throughout the network, it is assumed that each MAP has established symmetric shared keys with all its neighboring MAPs. After successfully authenticating C, M_1 encrypts I_C, I_{M1}, key K_{MAC}, and the pairwise master key PMK_0 shared with C by using the key M_1 shares with its neighbor M_x, and then M_1 sends the encrypted message to M_x. Upon receiving the message, M_x decrypts it and extracts K_{MAC} and PMK_0 to prepare for future authentication with C. The above computations are performed by MAPs, so there is no extra burden laid on the client's side. When C leaves M_1 and visits M_x, he/she executes the following handover authentication protocol:

1. Client C submits his/her transfer ticket θ_C, a new number $N_{C,}$ and Message authentication code $V_{KMAC}(N_C)$ to the foreign MAP M_x. Upon receiving this message, M_x verifies the correctness of $V_{KMAC}(N_C)$ by using K_{MAC} received from the home MAP M_1. If the verification turns out positive, M_x checks τ_{exp_θ} and the MAC value in θ_C to verify θ_C's validity. Of all clients, only C has the knowledge of K_{MAC} and can generate a valid pair (N_C, $V_{KMAC}(N_C)$). This enables the protocol to resist forgery attacks.
2. If the above verifications are successful, M_x sends a nonce N_M and a message authentication code $V_{KMAC}(N_C, N_M)$ to C. After checking out the received message, C produces a new pairwise master key $PMK_1 = f(PMK_0, N_C, N_M)$ for M_x. Then C sends N_M and $V_{KMAC}(N_M)$ to M_x to inform M_x he/she has successfully constructed PMK_1.
3. Upon receiving N_M and $V_{KMAC}(N_M)$, M_x verifies $V_{KMAC}(N_M)$. Since C is the only client that has K_{MAC}, a correct $V_{KMAC}(N_M)$ proves the identity of C. If the verification is successful, M_x also computes PMK_1. This concludes the handover authentication process.

4.3 Analysis of Li et al.'s authentication protocols

There are, unfortunately, some security flaws in Li et al.'s authentication protocols. For one, the expiration time and date of the transfer ticket θ_C are stored in plaintext. C can forge it and re-generate a matched MAC value to illegally extend validity. Secondly, the plaintext θ_C makes it possible for an adversary to track down a specific client, for θ_C involves I_C, and I_C is rarely changed. Thirdly, Li et al.'s protocols require that all MAPs should be equipped with high-quality TPDs so that the system can withstand physical attacks. In Li et al.'s design, MAP M_1 presends the same ticket to all its neighboring MAPs even though the client may probably move to only one of them (i.e. MAP M_2) but not the rest (i.e. MAP M_3 and MAP M_4). The latter MAPs, however, can still decrypt the ticket and obtain the same secret information (e.g., PMK). This increases security risks. Moreover, the domino effect [6] can also be a problem. The current pairwise master key (PMK) is generated by using the previous PMK along with some public information. Once some MAP M_n has obtained an old PMK, it can track down the current PMK at any time and even disguise as the client to communicate with other MAPs. In other words, once a MAP is compromised, all the MAPs directly or indirectly connected to it will also be affected, and so the whole security system will collapse.

5 The Proposed Authentication Protocols

To solve above problems, we shall present an efficient and secure authentication protocols as shown in Fig 4. Furthermore, the proposed protocols preserve clients' privacy well and only imposes negligible overhead. In the design of proposed protocols, TA is in charge of not only ticket issuing but also ECC public parameter management (e.g., $\{p, q, E(G), Z^*_n, P\}$ as in Section 3.3).

5.1 The Login Authentication Protocol (LAP)

As is stated in the preliminaries section, our login authentication protocol runs upon the assumption that the client and the MAP have respectively obtained a client ticket and a MAP

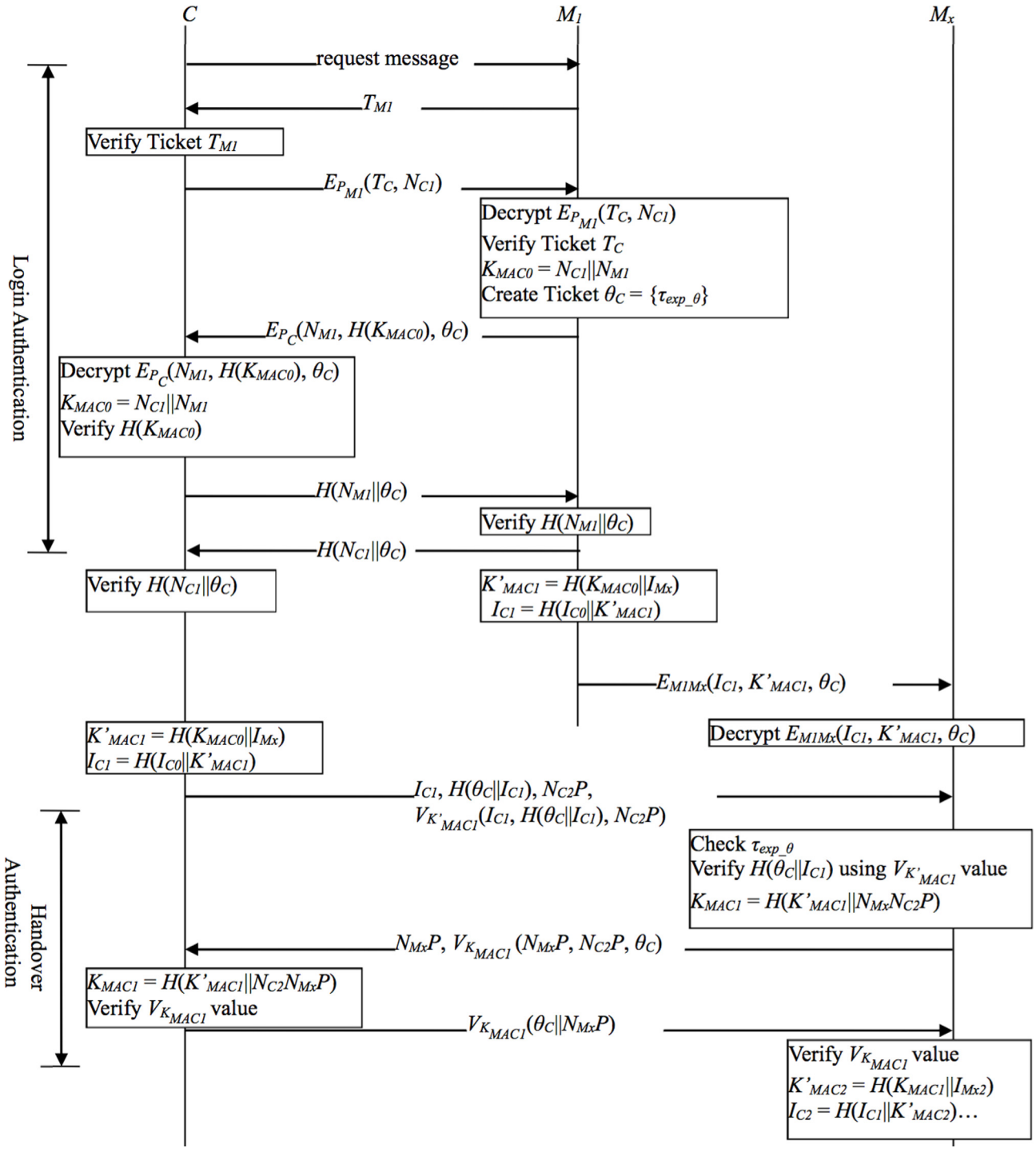

Fig 4. The proposed authentication protocols.

doi:10.1371/journal.pone.0155064.g004

ticket from TA. Now the client-MAP pair submit tickets to each other to do mutual authentication as follows.

1. When a client C starts to access the mesh network, he/she broadcasts a request message to the nearby MAPs.
2. Assuming that MAP M_1 replies with a message containing its MAP ticket T_{M1} to inform C of its presence. Upon receiving T_{M1}, C verifies Sig_A of T_{M1}. If Sig_A fails the verification, C ignores this T_{M1}; otherwise, C checks τ_{exp_M} of T_{M1}.
3. If the above verifications are successful, C extracts M_1's public key P_{M1} from T_{M1}. Then C encrypts his/her ticket T_C and a nonce N_{C1} using P_{M1}, and sends the encrypted message to M_1. Upon receiving the message, M_1 decrypts it and verifies Sig_A in T_C. If the verification fails, M_1 ignores T_C. Otherwise, M_1 checks τ_{exp_C} of T_C.
4. If the above verifications are successful, M_1 calculates the shared MAC key $K_{MAC0} = N_{C1}||N_{M1}$, creates a transfer ticket $\theta_C = \tau_{exp\_\theta}$, and extracts C's public key P_C from T_C, where N_{M1} is a nonce chosen by M_1. Then M_1 encrypts N_{M1}, $H(K_{MAC0})$, and θ_C using P_C, and sends the encrypted message to C. Upon receiving the message, C decrypts it to obtain N_{M1}, calculates the shared MAC key $K_{MAC0} = N_{C1}||N_{M1}$, and verifies $H(K_{MAC0})$. Note that we do not produce any extra pairwise master key PMK, so the bandwidth and key generation cost can both be reduced. In the proposed protocol, we set the MAC key or its derivatives as the session key between C and M_1. Thus, in the following steps, we only focus on the passing and using of the MAC key.
5. Then C sends $H(N_{M1}||\theta_C)$ to M_1 to prove that the decryption is successful. Upon receiving this message, M_1 checks to see if $H(N_{M1}||\theta_C)$ matches the value owned by itself. If the received $H(N_{M1}||\theta_C)$ does not check out, M_1 ignores this message.
6. To complete LAP, M_1 returns $H(N_{C1}||\theta_C)$ to C. Upon receiving this message, C checks to see whether the received $H(N_{C1}||\theta_C)$ matches the value owned by himself/herself. If the value does not match, C ignores this message.

This concludes LAP. The value K_{MAC0} will then be used to encrypt messages between C and M_1. On the other hand, C can use K_{MAC0} along with the transfer ticket θ_C to prove his/her legality and roam from M_1 to another MAP within the mash network.

5.2 The Handover Authentication Protocol (HAP)

To support fast handover for clients roaming from M_1 to another MAP, M_1 should pre-share some information to all its neighboring MAPs. Instead of broadcasting, we decide to take another route and construct a corresponding temporary MAC key $K'_{MAC1} = H(K_{MAC0}||I_{Mx})$ for every neighbor MAP M_x ($x = 2, 3, 4\ldots$). Due to the protection of the one-way hash function, even though M_x has its own K'_{MAC1}, it still cannot produce another MAP's K'_{MAC1}. In addition, to preserve clients' privacy and protect their identity information, we set $I_{C1} = H(I_{C0}||K'_{MAC1})$. That means every MAP M_x will obtain a number of anonymous ID numbers, one for a different client, and only this specific client C knows which anonymous ID number will be used. As is stated in the preliminaries section, it is assumed that each MAP has established symmetric keys with all its neighboring MAPs. After M_1 successfully authenticates the client C, M_1 encrypts the corresponding I_{C1}, K'_{MAC1}, and θ_C by using a different symmetric key shared with MAP M_x, and sends those encrypted messages to its neighboring MAPs. Note that M_1 sends the transfer ticket θ_C's original τ_{exp_θ} to M_x so that it will not be manipulated to illegally postpone the expiration time.

Upon receiving the message, M_x decrypts it using the same shared key and extracts K'_{MAC1} to prepare for future authentications with C. The above computations are performed by MAPs, and so no extra burdens are laid on the client. When C leaves M_1 and visits M_x, he/she executes the following handover authentication protocol:

1. Depending on the moving direction, C calculates a temporary MAC key $K'_{MAC1} = H(K_{MAC0}||I_{Mx})$ and $I_{C1} = H(I_{C0}||K'_{MAC1})$ in accordance with the ID number of the foreign MAP M_x to visit. Then, C chooses a nonce $N_{C2} \in Z^*_n$ and computes $N_{C2}P$. Note that C can prepare $N_{C2}P$ in advance so as to reduce the workload during HAP. If C knows which MAP is the next one yet to visit, he/she can also pre-compute I_{C1} and $H(\theta_C||I_{C1})$. Then, C submits I_{C1}, $H(\theta_C||I_{C1})$, $N_{C2}P$, and the MAC value $V_{K'MAC1}(I_{C1}, H(\theta_C||I_{C1}), N_{C2}P)$ to the foreign MAP M_x. Upon receiving the message, M_x checks the τ_{exp_θ} of θ_C received from M_1 and determines whether or not the τ_{exp_θ} has expired. Then, M_x verifies $H(\theta_C||I_{C1})$ and the MAC value by using K'_{MAC1}.
2. If the above verifications are successful, M_x chooses a nonce $N_{Mx} \in Z^*_n$, computes $N_{Mx}N_{C2}P$ and $N_{Mx}P$, and produces a formal MAC key $K_{MAC1} = H(K'_{MAC1}||N_{Mx}N_{C2}P)$. After that, M_x sends $N_{Mx}P$ and a MAC value $V_{KMAC1}(N_{Mx}P, N_{C2}P, \theta_C)$ to C. Upon receiving those messages, C computes $N_{C2}N_{Mx}P$ and also produces a formal MAC key $K_{MAC1} = H(K'_{MAC1}||N_{C2}N_{Mx}P)$ and verifies the MAC value transferred from M_x. If the verification result is positive, C sends $V_{KMAC1}(\theta_C||N_{Mx}P)$ to M_x to inform M_x he/she has successfully constructed K_{MAC1}.
3. Upon receiving $V_{KMAC1}(\theta_C||N_{Mx}P)$, M_x verifies the value. Since C is the only client that keeps K_{MAC1}, it proves the validity of C. Now the handover authentication process is completed, and K_{MAC1} is the session key used for further communications between C and M_x. Meanwhile, M_x can prepare K'_{MAC2} and I_{C2} for C's next execution of HAP.

6 Security Analysis

In this section, we list the common security requirements and threats [2, 9, 11–17, 22, 28, 30–46, 44–46], along with the proof that the proposed protocol can satisfy those requirements and withstand those threats. We shall compare with some similar protocols including Li et al.'s scheme [22] and Yang et al.'s scheme [45]. Yang et al.'s scheme is a follow-up study of Li et al.'s scheme. Yang et al. also found the ticket forgery problem in Li et al.'s scheme and proposed their solution in 2015. The security comparisons are shown as Table 2.

1) Mutual authentication:

Mutual authentication means that the participators of communication verify the legality of each other. In the proposed LAP, M_1 and C exchange and verify each other's ticket. The digital signature of TA can provide the legality of both M_1 and C. In addition, C encrypts his/her ticket and the authentication information by using M_1's public key. Under the protection of public

Table 2. Security comparison among similar protocols.

Security requirement	Li et al. [22]	Yang et al. [45]	Ours
Mutual authentication	Yes	Yes	Yes
Privacy preservation:	No	Yes	Yes
Forward and backward security	No	Yes	Yes
Replay attack resistance	Yes	Yes	Yes
Forgery attack resistance	No	Yes	Yes

doi:10.1371/journal.pone.0155064.t002

key cryptography, only M_1 can decrypt this message and extract the plaintext. Therefore, it is difficult for an attacker to obtain the authentication information (e.g., N_{C1}) and generate the correct reply message (e.g., $H(N_{C1})$). That means C can verify M_1's validity if M_1 returns correct $H(N_{C1})$. On the other hand, M_1 can also verify C if C returns correct $H(N_{M1})$, for only C can extract N_{M1} from $E_{PC}(N_{M1})$.

Before C moves from M_1 to M_x, M_1 encrypts K'_{MAC1} by using the symmetric key M_1M_x. Based on symmetric cryptography, M_1 and M_x can achieve mutual authentication with each other. Since the client only C has the necessary information for the construction of K'_{MAC1}, M_x can authenticate C by checking the MAC value based on K'_{MAC1} during HAP. In the meantime, C can authenticate M_x if M_x replies with the MAC value based on correct K_{MAC1}, for only M_x is capable of constructing the correct K_{MAC1}.

2) Privacy preservation:

In the proposed protocol, I_C is only used when C launches the login request, whereas I_{Ck}, which is for the k-th handover, is composed of a number of nonces provided by C and the MAPs that C roams through. In addition, I_C is encrypted by P_{M1} when the message is transmitted. That means two things: firstly, only M_1 can obtain I_C, and an interceptor cannot extract I_C from the encrypted message; secondly, M_1 cannot track I_{Ck} because it is always re-constructed by a couple of nonces provided respectively by C and the current MAP when C is roaming to a new MAP. Besides, instead of the plaintext θ_C, we use $H(\theta_C \,||I_{Ck})$ in HAP. That means it is hard to extract θ_C. Even when an adversary fortuitously obtains C's θ_C in plaintext, they still have to intercept all possible I_{Ck} and compute $H(\theta_C \,||I_{Ck})$ if he/she wish to track down C.

3) Forward and backward security:

To satisfy this requirement, we have to prove that, should an adversary somehow acquire a secret key for a certain communication session, the adversary has no way to derive the secret keys for the previous and following sessions. In the proposed scheme, the K_{MACk} for the k-th handover includes a couple of nonces respectively provided by C and the current MAP. Those nonces are protected by the elliptic curve discrete logarithm problem (ECDLP) and elliptic curve diffie-Helman (ECDH). That means given $\{N_{Mx}P, N_{Ck}P\} \in E(G)$, $\{N_{Mx}, N_{Ck}\} \in Z^*_n$, and $P \in G_q$, it is intractable to derive $N_{Mx}N_{Ck}P$. Therefore, even if $N_{Mx}P$ and $N_{Ck}P$ are intercepted by an adversary, the adversary still cannot derive the MAC key $N_{Mx}N_{Ck}P$. Furthermore, those nonces are randomly generated, so an adversary cannot derive a new MAC key should the adversary be able to break an old MAC key. In addition, since the k-th secret key is constructed from the collision-free one-way hash function of the (k-*1*)-th secret key, an adversary cannot derive any previous secret keys from it. To conclude, the proposed protocol guarantees perfect forward /backward security.

4) Replay attack resistance:

An adversary may eavesdrop some messages during authentication sessions and replay these messages in the future in an attempt to get authenticated and successfully access the network as a client. Similarly, an attacker may attempt to gain the client's trust as a MAP. To protect clients and MAPs from reply attacks, we set two nonces in LAP and HAP. In LAP, the nonces N_{C1} and N_{M1} are randomly generated by C and M_1 respectively and protected by public key cryptography. An adversary cannot succeed in being authenticated by replaying the message because C and M_1 discard those nonces once the LAP is completed. Besides, since we use public key cryptography and one-way hash function to protect the nonces N_{C1} and N_{M1}, the adversary cannot decrypt the messages and thus cannot launch a successful replay attack when the LAP is ongoing.

In HAP, the private MAC key and nonces protect the client and MAP from replay attacks. What the attacker has in hand are only the old nonces and an old MAC value recorded, which cannot pass a new authentication procedure. Even if the attacker should come by some new nonces, there is still no way they can compute the correct MAC value in the absence of the private MAC key. In conclusion, both our LAP and HAP can resist the replay attack.

5) Forgery attack resistance:

In LAP, TA's digital signature ensures that both the client's ticket and the MAP's ticket are valid and unmodified. In HAP, since the MAP that C is leaving M_1 for has obtained the correct information of θ_C from M_1, C cannot forge a fabricated θ_C. On the other hand, since the MAC value is different for each round-trip, we can ensure the integrity of these messages and thus guarantee that our HAP can resist attacks from outside. Any unofficial modification to the content of message will result in an incorrect MAC value due to the lack of the MAC key and will be identified.

7 Performance Analysis

In this section, we shall analyze the performance of the proposed handover authentication scheme by showing how it compares with some similar protocols including EAP-TLS [47], Li et al.'s scheme [22], and Yang et al.'s scheme [45]. EAP-TLS is a popular authentication protocol for IEEE 802.11-based wireless networks and represents the multi-hop handover authentication approach. Note that to keep real-time applications going and to provide better user experience, the overall handover latency should not exceed 50ms [2].

7.1 Computation Cost

The computation cost represents the processing delays of the cryptographic operations on both the client side and the MAP side. These operations include encryption using public key (T_E), decryption using private key (T_D), generation of digital signature (T_{sig}), verification of digital signature (T_{ver}), computation of MAC value (T_{MAC}), computation of hash value(T_H), and computation of point multiplication (T_{pmul}). To be fair, we used the same public key cryptographic system—RSA-1024 as the RFC 5216 suggested for T_E, T_D, T_{sig}, and T_{ver}.—to run all the schemes compared. We learned the authentication latencies of T_E, T_D, T_{sig}, T_{ver}, T_{MAC}, and T_H from Long and Wu's experimental results [48]. However, Long and Wu's experimental results do not include the latency of T_{pmul}. Although they did provide ECDSA's signature latencies, those operations are way too complex, so the results cannot translate to the latency of T_{pmul}. Therefore, we turned and referred to other studies concerned to obtain the ratio of ECC to RSA. Since the decryption of ECC only involves a point multiplication and a point subtraction, we can say that the cost of time for the decryption of ECC is almost the same as that of T_{pmul}. According to Ariffin and Mahad's study, the time cost of decryption at a ratio of ECC-128 to RSA-1024 is approximately 0.0113 (0.770s: 68.042s per 625 blocks) [49]. We can then translate this to an approximated T_{pmul} time cost of 0.376ms (33.3ms×0.0113). Please note that in Long and Wu's experiment RSA used a short-length public key and a long-length privacy key to expedite the process of public key operations (e.g., T_E and T_{ver}), and the ratio of T_D to T_E was approximately 23.45. Because we use the public key operations as the intermediary value, the time cost of T_{pmul} may be overestimated. To clearly see how the schemes compared in terms of performance, Table 3 lists the operations discussed above, the algorithms implementing the operations, time cost of these algorithms, and the numbers of different kinds of operations performed by different protocols.

Table 3. Performance comparison among similar protocols.

Op. (Algorithm)	Time(*ms*)	EAP-TLS	Li et al.		Yang et al.		Ours	
			LAP	HAP	LAP	HAP	LAP	HAP
T_E (RSA-1024)	1.420	1	2	0	2	0	2	0
T_D (RSA-1024)	33.30	1	2	0	2	0	2	0
T_{sig} (RSA-1024)	33.30	1	0	0	1	0	0	0
T_{ver} (RSA-1024)	1.420	3	2	0	2	1	2	0
T_{MAC} (HMAC)	0.015	0	1	6	1	4	0	5
T_H (SHA-1)	0.009	3	0	0	0	0	6	3
T_{pmul} (ECC-128)	$\approx$ 0.376	0	0	0	0	0	0	2
Total computation cost (*ms*)		72.307	72.295	0.090	105.595	1.480	72.334	0.854
Number of transmissions		9	6	3	5	3	6	3
Authentication latency (*ms*)		72.307+9*dh*	72.295+6*d*	0.09+3*d*	105.595+5*d*	1.48+3*d*	72.334+6*d*	0.854+3*d*

doi:10.1371/journal.pone.0155064.t003

Because the first T_{pmul} can be performed in advance (e.g., N_MP and $N_{C2}P$), the number of T_{pmul} is cut down from two to one for the client. Similarity, the MAP also only needs to perform one T_{pmul}. That means the total number of T_{pmul} is only two in HAP. According to Table 2, the total computation cost of the proposed scheme in LAP (72.334*ms*) is slightly higher than those of EAP-TLS (72.307*ms*) and of Li et al. (72.259*ms*) but significantly lower than Yang et al.'s scheme (1*05.595ms*). However, since the client only needs to do LAP one time and it lasts until the transfer ticket expires, the impact of a long LAP processing time is relatively small. More importantly, the computation cost of the proposed scheme in HAP is only 0.854*ms* and significant lower than Yang et al.'s scheme. Although the proposed HAP is slower than Li et al.'s scheme, the price is paid to mend the security flaws in Li et al.'s protocols mentioned earlier, and we think it is worth the while. With the overall handover latency kept under 1*ms*, the proposed scheme still shows pretty good efficiency.

7.2 Communication Cost

The communication cost is estimated in accordance with the number of message transmissions between a MAP and a client during LAP or MAP because the transmissions are what cause the communication delays. The notation *d* stands for the average delay of a transmission made across one hop, and *h* is the number of hops between the client and the verifier. Because EAP-TLS is the only scheme in the comparison that requires a client to communicate with the EAP server, which is always multi-hops away, during the handover process, the parameter *h* is applicable to only EAP-TLS. The results show that the number of transmissions made between the client and the MAP in the proposed HAP is equal to those of Li et al.'s and Yang et al.'s and is smaller than EAP-TLS. In other words, judging by communication cost, the proposed scheme performs just as well as Li et al.'s scheme.

8 Conclusion

In this paper, we have proposed fast handover authentication protocols based on ticket for wireless mesh network. Not only does the proposed protocol satisfy all the essential handover security requirements, but it also preserves the privacy of the client. The results of our security analysis and performance comparison show that the proposed protocol is superior to the other protocols of the kind. Even though the proposed HAP is slightly slower than Li et al.'s work, the security level is significantly higher. Moreover, like the proposed protocol, Yang et al.'s protocol is also the product of a follow-up study meant to fix the security problems of Li et al.'s

work, but our new protocol turns out to be both faster and more secure. Besides, in our new design, no complicated computations need to be done on the client's side, and therefore our new protocol is especially suitable for applications in wireless mesh networks where the mobile devices have limited computation power. Our future research will continue to focus on the development of authentication protocols that are more efficient, more reliable, and more user-friendly.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This research was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract no.: MOST 104-2221-E-030-002. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author Contributions

Conceived and designed the experiments: YML PJC CCL CYK. Performed the experiments: YML PJC CCL CYK. Analyzed the data: YML PJC CCL CYK. Contributed reagents/materials/analysis tools: YML PJC CCL CYK. Wrote the paper: YML PJC CCL CYK.

References

1. Guo P, Wang J, Li B, Lee S. A variable threshold-value authentication architecture for wireless mesh networks. Journal of Internet Technology. 2014; 15(6):929–36.
2. International Telegraph and Telephone Consultative Committee. General characteristics of international telephone connections and circuits: recommendations G.101-G.181. *International Telecommunication Union*; 1985.
3. Srivatsa AM, Xie J. A performance study of mobile handoff delay in IEEE 802.11-based wireless mesh networks. Proceedings of *IEEE International Conference on Communications*; 2008 May 1923; Beijing, China: IEEE; 2008. p. 2485–2489.
4. Blom R. An optimal class of symmetric key generation systems. Advances in Cryptology. 1985; 209:335–8.
5. Du W, Deng J, Han YS, Varshney PK, Katz J, Khalili A. A pairwise key predistribution scheme for wireless sensor networks. ACM Transactions on Information and System Security. 2005; 8(2):228–58.
6. Fu A, Lan S, Huang B, Zhu Z, Zhang Y. A novel group-based handover authentication scheme with privacy preservation for mobile WiMAX networks. IEEE Communications Letters. 2012; 16(11):1744–7.
7. Fu A, Zhang G, Zhang Y, Zhu Z. GHAP: An efficient group-based handover authentication mechanism for IEEE 802.16m networks. Wireless Personal Communications. 2013; 70(4):1793–810.
8. Fu A, Zhang G, Zhu Z, Zhang Y. Fast and secure handover authentication scheme based on ticket for WiMAX and WiFi Heterogeneous networks. Wireless Personal Communications. 2014; 79(2):1277–99.
9. Fu A, Zhang Y, Zhu Z, Jing Q, Feng J. An efficient handover authentication scheme with privacy preservation for IEEE 802.16 m network. Computers & Security. 2012; 31(6):741–9.
10. Fu A, Zhang Y, Zhu Z, Liu X. A fast handover authentication mechanism based on ticket for IEEE 802.16m. IEEE Communications Letters. 2010; 14(12):1134–6.
11. Jiang Q, Ma J, Li G, Yang L. An efficient ticket based authentication protocol with unlinkability for wireless access network. Wireless Personal Communications. 2014; 77(2):1489–506.
12. Jiang Q, Ma J, Li G, Yang L. An enhanced authentication scheme with privacy preservation for roaming service in global mobility networks. Wireless Personal Communications. 2013; 68(4):1477–91.
13. Qi Jing, Zhang Y, Fu A, Liu X. A privacy preserving handover authentication scheme for EAP-based wireless networks. Proceedings of *IEEE Global Telecommunications Conference*; 2011 Dec 05–09; Houston, TX, USA: IEEE; 2011. p. 1–6.
14. Khedr WI, Abdalla MI, Elsheikh AA. Enhanced inter-access service network handover authentication scheme for IEEE 802.16 m network. IET Information Security. 2015; 9(6):334–43.

15. Kohl J, Neuman C. The Kerberos network authentication service (V5). Network Working Group, RFC 1510, 1993.
16. Li G, Chen X, Ma J. A ticket-based re-authentication scheme for fast handover in wireless local area networks. Proceedings of *2010 6th International Conference on Wireless Communications Networking and Mobile Computing*; 2010 Sep 23–25; Chengdu, China: IEEE; 2010. p. 1–4.
17. Li G, Ma J, Jiang Q, Chen X. A novel re-authentication scheme based on tickets in wireless local area networks. Journal of Parallel and Distributed Computing. 2011; 71(7):906–14.
18. Tie L, Yi Y. Extended security analysis of multi-hop ticket based handover authentication protocol in the 802.16j network. Proceedings of *2012 8th International Conference on Wireless Communications Networking and Mobile Computing*, *Shanghai*; 2012 Sep 21–23; Shanghai, China: IEEE; 2012. p. 1–10.
19. Wienzek R, Persaud R. Fast re-authentication for handovers in wireless communication networks. Proceedings of *5th International IFIP-TC6 Networking Conference*; 2016 May 15–19; Coimbra, Portugal: Springer; 2006. p. 556–567.
20. Xie Q, Hu B, Dong N, Wong DS. Anonymous three-party password-authenticated key exchange scheme for telecare medical information systems. PLoS ONE. 2014; 9(7): e102747. doi: 10.1371/journal.pone.0102747 PMID: 25047235
21. Xu L, He Y, Xiaofeng C, Huang X. Ticket-based handoff authentication for wireless mesh networks. Computer Networks. 2014; 73:185–94.
22. Li C, Nguyen UT, Nguyen HL, Huda N. Efficient authentication for fast handover in wireless mesh networks. Computers & Security. 2013; 37:124–42.
23. Sohn I. Access point selection game with mobile users using correlated equilibrium. PLoS ONE. 2015; 10(3): e0116592. doi: 10.1371/journal.pone.0116592 PMID: 25785726
24. Sun J, Zhang C, Zhang Y, Fang Y. An identity-based security system for user privacy in vehicular Ad hoc networks. IEEE Transactions on Parallel and Distributed Systems. 2010; 21(9):1227–39.
25. Krawczyk H, Bellare M, Canetti R. HMAC: keyed-hashing for message authentication. Network Working Group, RFC 2104, 1997.
26. Manuel S. Classification and generation of disturbance vectors for collision attacks against SHA-1. Designs, Codes and Cryptography. 2011; 59(3):247–63.
27. Miller VS. Use of elliptic curves in cryptography. Advances in Cryptology. 1985; 218:417–26.
28. Jiang Q, Khan MK, Lu X, Ma J, He D. A privacy preserving three-factor authentication protocol for e-Health clouds. The Journal of Supercomputing. 2016; doi: 10.1007/s11227-015-1610-x
29. Lee CC, Chiu ST, Li CT. Improving security of a communication-efficient three-party password authentication key exchange protocol. International Journal of Network Security. 2015; 17(1):1–6.
30. Chuang MC, Lee JF. SF-PMIPv6: A secure fast handover mechanism for Proxy Mobile IPv6 networks. Journal of Systems and Software. 2013; 86(2):437–48.
31. Chuang MC, Lee JF, Chen MC. SPAM: a secure password authentication mechanism for seamless handover in proxy mobile IPv6 networks. IEEE Systems Journal. 2013; 7(1):102–13.
32. He D, Kumar N, Chen J, Lee CC, Chilamkurti N, Yeo SS. Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. Multimedia Systems. 2015; 21 (1):49–60.
33. He D, Kumar N, Chilamkurti N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. Information Sciences. 2015; 321:263–77.
34. He D, Zeadally S. Authentication protocol for ambient assisted living system. IEEE Communications Magazine.2015; 35(1):71–7.
35. He D, Wang D. Robust biometrics-based authentication scheme for multi-server environment. IEEE Systems Journal. 2015; 9(3):816–23.
36. Islam SKH, Khan MK, Al-Khouri AM. Anonymous and provably secure certificateless multireceiver encryption without bilinear pairing. Security and Communication Networks. 2015; 8(13):2214–31.
37. Jiang Q, Ma J, Lu X, Tian Y. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. Peer-to-Peer Networking and Applications. 2015; 8(6):1070–81.
38. Jiang Q, Wei F, Fu S, Ma J, Li G, Alelaiwi A. Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy. Nonlinear Dynamics. 2016; 83(4):2085–101.
39. Khan MK. Fingerprint biometric-based self and deniable authentication schemes for the electronic world. IETE Technical Review. 2009; 26(3):191–5.
40. Khan MK, Zhang J, Lei Tian. Protecting biometric data for personal identification. Advances in Biometric Person Authentication. 2004; 3338:629–38.

41. Khan MK, Zhang J. An efficient and practical fingerprint-based remote user authentication scheme with smart cards. Information Security Practice and Experience. 2006; 3903:260–8.
42. Lee CC, Li CT, Chiu ST, Lai YM. A new three-party authenticated key agreement scheme based on chaotic maps without password table. Nonlinear Dynamics. 2015; 79(4):2485–95.
43. Lee CC, Li CT, Hsu CW. A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps. Nonlinear Dynamics. 2013; 73(1):125–32.
44. Qazi S, Mu Y, Susilo W. Securing wireless mesh networks with ticket-based authentication. Proceedings of *2nd International Conference on Signal Processing and Communication Systems*; 2008 Dec 15–17; Gold Coast, Australia: IEEE; 2008. p. 1–10.
45. Yang X, Huang X, Han J, Su C. Improved handover authentication and key pre-distribution for wireless mesh networks. Concurrency and Computation: Practice and Experience. 2015; doi: 10.1002/cpe.3544
46. Yoon EJ, Yoo KY. A forgery attack on a low computation cost user authentication scheme. International Journal of Network Security. 2006; 3(1):51–3.
47. Simon D, Aboba B, Hurst R. The EAP-TLS authentication protocol. Network Working Group, RFC 5216, 2008.
48. Long M, Wu CHJ. Energy-efficient and intrusion-resilient authentication for ubiquitous access to factory floor information. IEEE Transactions on Industrial Informatics. 2006; 2(1):40–7.
49. Ariffin MRK, Mahad Z. AAβ public key cryptosystem-a comparative analysis against RSA and ECC. Proceedings of 2012 7th International Conference on Computing and Convergence Technology: 2012 Dec 03–05; Seoul, Korea: IEEE; 2012. p. 589–94.