An IEEE 802.11 Fast Reassociation and Pairwise Transient Key establishment Based on the Dynamic Cluster Method

Chung-Ming Huang and Jian-Wei Li
Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan, Taiwan 701, R.O.C
huangcm@locust.csie.ncku.edu.tw

ABSTRACT

Wireless local area network (WLAN) based on the IEEE 802.11 technology becomes popular in recent years. However, the latency of the handoff process in WLAN is very large such that large handoff gap may bring excessive jitter in multimedia applications, e.g., voice over IP. Therefore, many researches have made efforts on how to fast handoff. In this paper, we propose an accelerated handoff mechanism in which two methods are involved: (1) Pairwise master key security association (PMKSA) distribution based on the dynamic cluster selection and transition and (2) fast reassociation with the pairwise transient key security association (PTKSA) establishment. Access points (APs) that are cluster members can cache PMKSA of client stations (STAs) in advance to reduce the 802.1X authentication delay. The former method is proposed to assure that STA stays within a cluster. In the latter, the fast reassociation with the PTKSA establishment process incorporates four-way-handshake into the IEEE 802.11 reassociation process to actively raise handoff process and to further accelerate handoff process.

***Keywords**: authentication, BSS transition, cluster, cluster roaming key, handoff, WLAN*

1: INTRODUCTIONS

The IEEE 802.11 handoff in wireless local area network (WLAN) can be divided into the following scenarios: (1) No transition: A client station (STA) moves within a basic service set (BSS). (2) BSS transition: STA moves from one AP to another within the same extended service set (ESS). (3) ESS transition: STA moves from a BSS in one ESS to another BSS in a different ESS. In this paper, we study the IEEE 802.11 handoff in terms of BSS transition.

If STA wants to associate with an access point (AP) for connecting to Internet, STA must perform a full AP contact process with the AP, which purpose is to determine an AP, then associate and authenticate with the AP. In the original IEEE 802.11 standard [2], the full AP contact process only contains probe process, open authentication process and (re)association process. Due to the special property of IEEE 802.11 technology that any piece of message is transmitted through the open air, it is an important issue to prevent transmitted messages from being intercepted. The Wired Equivalent Privacy (WEP) mechanism was proposed to protect transmitted messages. Nevertheless, many studies have demonstrated that the WEP is insecure [3]. Afterward, to enhance the security of IEEE 802.11 technology, IEEE 802.11i specifies the IEEE 802.1X network access control mechanism and defines a four-way handshake protocol [5]. Therefore, in the IEEE 802.11i standard, the full AP contact process contains (1) probe process, which determines an AP from available APs as the next associated AP that is called the target AP, (2) (re)association process, (3) IEEE 802.1X access control mechanism, and (4) four-way-handshake protocol.

In WLAN, when moving from one AP to another, STA also needs to perform the full AP contact process. Since STA can associate with no more than one AP, STA must de-associate with the visited AP before re-associating with the target AP. STA can not receive any data frame from the previous visited AP during the period from de-associating with the previous visited AP to re-associating successfully with the target AP, in which the period is called handoff gap. In Shin et al.'s study [11], probe delay is bout 380ms, which is about 90% overhead of the full AP contact process in the original IEEE 802.11 standard [2]. Therefore, a great deal of effort has been made on probe delay reduction [10,12]. However, if IEEE 802.11i is considered, in which the delay is about 120ms, the latency of IEEE 802.11i brings a large handoff gap. Since the bearable maximum delay is 50ms in multimedia applications [6], such a large handoff gap may bring excessive jitter. Therefore, many researches have made efforts on how to shorten the handoff latency.

In 2004, Pack and Choi have proposed a fast inter-AP handoff scheme with a predictive authentication based on a frequent handoff region (FHR) [9]. However, Pack and Choi did not consider IEEE 802.11i. Afterward, Mishra et al. considered IEEE 802.11i and proposed a pre-authentication scheme based on proactive key distribution using a neighbor graph, which is adopted by IEEE 802.11f (Inter-Access Point Protocol; IAPP) [4,7,8]. The neighbor graph dynamically captures the topology of the network to trace the potential APs which STA may handoff to in the next time. However, Mishra et al.'s scheme, each time when handoff process is done, the visited AP must distribute pairwise master key security association (PMKSA) information about STA to its neighboring

The research is supported by the National Science Council of the Republic of China under the grant number NSC 95-2219-E-006-008 and the Program of Top 100 Universities Advancement, Ministry of Education, Taiwan, Republic of China.

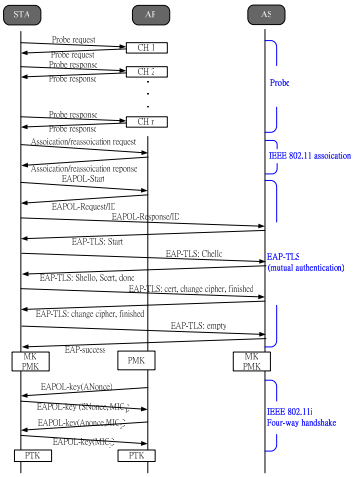

Figure 1. The full AP contact procedure in WLAN.

APs that are determined according to the neighbor graph. In other words, if one STA has performed *n* handoff processes, the *n* PMKSA information distribution to neighboring APs also must be involved. It brings about certain of bandwidth consuming in ESS. Furthermore, since Mishra et al.'s scheme did not take account of the latency of four-way handshake of IEEE 802.11i in the handoff gap. Since four-way handshake needs four message exchanges between STA and AP, the handoff gap can be further reduced if the four-way handshake is optimized. Furthermore, since STA needs to retrieve a nonce from AP in four-way handshake, STA can not actively raise four-way handshake on demand.

In this paper, we propose an accelerated handoff mechanism, in which two methods are involved: (1) PMKSA distribution based on a dynamic cluster selection and transition and (2) fast reassociation with the PTKSA establishment. After STA has associated with the visited AP using the full AP contact process, the visited AP selects a cluster and determines the cluster members according to a neighbor graph. These APs that are the cluster members can cache PMKSA of STA in advance. Since PMKSA of STA has existed, the extensible authentication protocol- transport Layer Security (EAP-TLS) authentication delay can be eliminated if STA handoffs to an AP that is the cluster member [1]. The dynamic cluster selection and transition method is proposed to assure that STA stays within a cluster. Once PMKSA has existed, STA can thus directly perform a fast reassociation with the pairwise transient key security association (PTKSA) establishment process to re-associate with the target AP and agree on a pairwise transient key (PTK), where the fast reassociation with the PTKSA establishment process incorporates four-way-handshake into the IEEE 802.11 reassociation process. Therefore, PTKSA is simultaneously established during the IEEE association process.

2: RELATED WORKS

In this section, we briefly review IEEE 802.11i standard and the neighbor graph.

2.1: IEEE 802.11i

IEEE 802.11i Task Group I defines a security enhancement framework for IEEE 802.11. Here, we briefly review IEEE 802.1X network access control mechanism and four-way-handshake protocol in IEEE 802.11i.

IEEE 802.1X network access control As Figure 1 depicted, IEEE 802.11i standard specifies IEEE 802.1X as access control framework, which is a port-based access control in the link layer. The port corresponds to an association between AP and STA in the IEEE 802.11 scenario. IEEE 802.1X standard employs EAP to permit a variety of authentication schemes, where IEEE802.11i standard specifies the EAP-TLS as the authentication scheme. After EAP-TLS completes, Authentication Server (AS) and STA authenticate with each other and agree a master key (MK) and a pairwise master key (PMK). Next, AS distributes the PMK to the visited AP. Thus, STA and the visited AP have the PMK, i.e., STA and the visited AP has installed PMKSA.

Four-Way handshake The communication in four-way handshake is carried using the EAP over LAN (EAPOL) messages. The four-way handshake is used to guarantee the freshness and synchronizes the shared session key, i.e., PTK, between AP and STA. After four-way handshake completes, STA and the visited AP have the PTK, i.e., STA and the visited AP have installed PTKSA. The four-way handshake is as follows:

AP→STA: The AP sends message M_a which an EAPOL-key message together with a nonce *ANonce* generated by AP to STA.

STA→AP: Upon receipt of M_a, STA generates a nonce *SNonce* and then can derive *PTK* as

$$PTK = PRF(PMK, ANonce \mid SNonce \mid MAC_{ap} \mid MAC_s) \quad (1)$$

where *PRF*(.)is a pseudo-random function [5]. MAC_{ap} and MAC_s stand for the MAC addresses of AP and STA, respectively. Then, *PTK* is divided into three keys. (i) EAPOL-Key confirmation key (*KCK*), which is used to provide EAPOL-key frame data integrity check and origin authenticity. (ii) EAPOL-Key encryption key (*KEK*), which is used to distribute group temporal key and keep its confidentiality. (iii) temporal key, which is used to secure data traffic. After that, STA sends M_b to the visited AP. M_b is an EAPOL-key message with *SNonce* and message integrity code (MIC_1) which is over M_b to protect its integrity.

AP→STA: Upon receipt of M_b, AP can derive *PTK* using Equation (1). Then AP can use *KCK* to verify MIC_1. If valid, AP sends M_c to STA, where M_c includes an EAPOL-key message with *ANonce* and MIC_2.

STA→AP: Upon receipt of M_c, STA verifies MIC_2 retrieved from M_c. If the result is positive, STA sends M_d to the visited AP, where M_d includes an EAPOL-key message together with *ANonce* and MIC_3. Similarly, upon receipt of M_d, the visited AP can verify MIC_3 to check its data integrity and origin authenticity.

2.2: NEIGHBOR GRAPH

The neighbor graph is constructed at each AP, where $NG(ap_i)$ stands for the neighbor graph at AP ap_i [8]. The

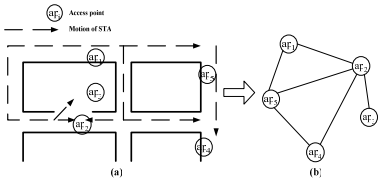

Figure 2. An illustrated neighbor graph, (a) a physic topology of WLAN and (b) corresponding neighbor graph [8]

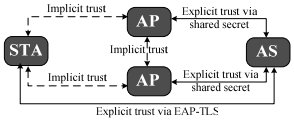

Figure 3. Trust relationships.

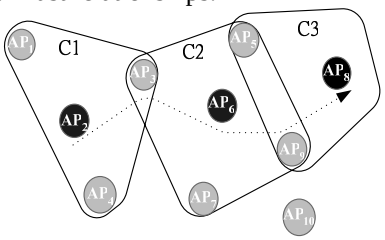

Figure 4. PMKSA distribution based on dynamic cluster selection and transition

neighbor graph is constructed by defining an undirected graph $G = (V, E)$, where $V = \{ap_1, ap_2, ... ap_m\}$ is the set of all APs and there is an edge $E = (ap_i, ap_j)$ between ap_i and ap_j if they satisfy a reassociation relationship. The reassociation relationship means that STA can perform an 802.11 reassociation through some path of motion between the physical locations of ap_i and ap_j. As Figure 2-(a) depicted, since STA has path in physic topology from ap_1 to ap_2 and ap_5, the edges (ap_1, ap_2) and (ap_1, ap_5) satisfy the reassociation relationship. As a result of satisfying the reassociation relationship, the edges (ap_1, ap_2) and (ap_1, ap_5) exist in the neighbor graph depicted in Figure 2-(b). According the definition of neighbor graph, the neighbor graph can be constructed as Figure 2-(b).

3: THE ACCELERATED HANDOFF MECHANISM

In the proposed mechanism, two methods are involved: (1) PMKSA distribution based on dynamic cluster selection and transition and (2) fast reassociation with the PTKSA establishment. Both methods work under the trust relationships as depicted in Figure 3. In the proposed mechanism, trust relationships exist among STA, APs and AS. As Figure 3 depicted, in an ESS, STA and AS can use EAP-TLS authentication scheme to explicitly trust each other. AS and AP use a shared secret to explicitly trust each other. Therefore, STA and AP can retrieve some shared secret, i.e., *PMK*, via both explicit trusts to implicitly trust each other, which is specified in IEEE 802.11i standard [5]. APs can retrieve some shared secret via explicit trust between AP and AS to implicit trust each other, which is specified in IEEE 802.11f standard, i.e., IAPP [4].

3.1: PMKSA DISTRIBUTION BASED ON DYNAMIC CLUSTER SELECTION AND TRANSITION

The dynamic cluster selection and transition method is to select a cluster for STA, and dynamically re-select and transit to a new one. Once STA has associated with the visited AP using the full AP contact process depicted in Figure 1, the visited AP ap_i selects the first cluster C_1 and determines the cluster members $CM(C_1)$ according to the neighbor graph, where the visited AP ap_i is also called the center AP of cluster. Besides, STA retrieves the list of member APs of C_1 (*lcm*) from the center AP in C_1. However, STA may handoff to APs that are not in $CM(C_1)$. Therefore, once STA determines to handoff to the first of AP which is not in $CM(C_1)$, the selected cluster must be re-selected, which is marked as the 2nd cluster C_2, and be transited to it. Similarly, STA may move from AP in $CM(C_{r-1})$ to AP in $CM(C_r)$. As Figure 4 depicted, the cluster is dynamically reselected and transited to it along the movement of STA. The black circular point ap_2, ap_6 and ap_8 are the center of clusters. That is, the selected cluster is re-selected in the order of ap_2, ap_6 and ap_8. As figure 4 depicted, since the cluster can be dynamically selected along the movement of STA and be transited to it, STA can be assured to have more probability of staying in the cluster.

In the cluster C_1, the center AP of C_1 can generates a cluster roaming key (*CRK*) as

$$CRK=PRF(PMK_i,MAC_s|lcm) \quad (2)$$

where PMK_i is retrieved from the AP contact process, MAC_s is the MAC address of STA and *lcm* denotes the list of APs $\in CM(C_1)$, and then distributes *CRK* to APs $\in CM(C_1)$. Since APs $ap_h \in CM(C_1)$ can derive PMK as

$$PMK_h = PRF(CRK, MAC_s \mid MAC_{ap_h}), \quad (3)$$

these APs have PMKSA of STA. In other words, the center AP of C_1 distributes PMKSA of STA to APs $\in CM(C_1)$. However, when STA transits to C_2 or succeeding clusters, the center APs of these clusters do not have PMKSA information of STA. One way is that the center APs can request PMKSA information of STA from an AP which STA previously visited to. Upon retrieving PMKSA information, the center APs can distribute PMKSA of STA to their cluster members.

3.2: THE FAST REASSOCIATION WITH THE PTKSA ESTABLISHMENT PROCESS

The IEEE 802.11 reassociation process and four-way handshake in IEEE 802.11i standard need 6 message exchanges in total to achieve the reassociation and PTKSA establishment between AP and STA. To shorten the handoff gap, a fast reassociation with the PTKSA establishment process is proposed, which incorporates the four-way-handshake into the IEEE 802.11 reassociation process.

Self nonce generation In the first and second message exchanging of four-way-handshake, both AP and STA need to generate random numbers, which are called nonces, and exchange messages with each other to agree

Algorithm 1 Fast reassociation with the PTKSA establishment process on the STA side

Require:
ap_j is the target AP
if $Ssyn$ exists for ap_j **then**
  retrieve $Ssyn$ from its database
else
  create $Ssyn$ for ap_j
  $Ssyn \leftarrow 1$
end if
repeat
  Generate $SNonce$
  $Derive\_ANonce(PMK_j, MAC_{ap_j}, MAC_s, Ssyn)$
  $Derive\_PTK_j(PMK_j, MAC_{ap_j}, MAC_s)$
  Generate MIC_s
  $Send\_ReassoREQ(ap_j, sp)$
  Wait until $Receive\_ReassoREP(ap_j, flag)$
  if flag is G **then**
    Verify MIC_a
    if valid **then**
      $Ssyn \leftarrow Ssyn + 1$
    end if
  else if flag is A **then**
    $Ssyn \leftarrow Asyn$
  end if
until received reassociation reply indicates grant
restore $Ssyn$ to database

Algorithm 2 Fast reassociation with the PTKSA establishment process on the AP side

Require:
s is STA
$Recieve\_ReassoREQ(s, sp)$
Confirm whether $Asyn$ for s exists or not.
if exist **then**
  retrieve $Asyn$ from its database
else
  create $Asyn$ for s
  $Asyn \leftarrow 1$
end if
Compare $Ssyn? = Asyn$
if equal **then**
  $Verify\_ANonce(PMK_j, MAC_{ap_j}, MAC_s, Asyn)$
  if equal **then**
    $Derive\_PTK_j(PMK_j, MAC_{ap_j}, MAC_s)$
    Verify MIC_s
    if valid **then**
      Generate MIC_a
      Open a port for STA
      $Send\_ReassoREP(ap_j, G)$
    else
      $Send\_ReassoREP(s, R)$
    end if
  else
    $Send\_ReassoREP(ap_j, A)$
  end if
else
  $Send\_ReassoREP(ap_j, A)$
end if

on PTK. To eliminate nonce exchange and be raised actively by STA, we propose a self nonce generation scheme. STA can generate nonce *SNonce* and *ANonce* by itself. In the self nonce generation scheme, *ANonce* for AP ap_j is derived as

$$ANonce = PRF(PMK_j, MAC_{ap_j} | MAC_s | Ssyn) \quad (4)$$

where *Ssyn* means a parameter on STA for synchronization. On the other hand, there is a parameter on AP for synchronization, called *Asyn*. When STA (re)associates with AP, two synchronization parameters *Ssyn* and *Asyn*, whose values should be the same and initial value is 1, are created for STA and AP, respectively.

Operations of the process In this section, Algorithm 1 and Algorithm 2 are defined to deal with that STA and the target AP perform the fast reassociation with the PTKSA establishment process, respectively. In Algorithm 1 and Algorithm 2, the following functions are involved.

- $Derive\_ANonce(PMK_j, MAC_{ap_j}, MAC_s, Ssyn)$: STA *s* calculates *ANonce* of the visited AP ap_j using Equation (4).
- $Derive\_PTK_j(PMK_j, MAC_{ap_j}, MAC_s)$: STA *s* derives PTK_j as

$$PTK_j = PRF(PMK_j, ANonce | SNonce | MACap_j | MAC_s) \quad (5)$$

- *Send_ReassoREQ(*ap_j*,sp)* and *Recieve_ReassoREQ(s,sp)*: The former means sending a reassociation request to AP ap_j together with security parameter (*sp*), including *Ssyn*, MIC_s, *SNonce* and *ANonce*, etc. The latter means receiving a reassociation request from STA *s*.
- *Send_ReassoREP(s,flag)* and *Recieve_ReassoREP(*ap_j*,flag)*: The former function means that AP ap_j sends a reassociation reply to STA *s* together with a flag which is set as G, A or R meaning grant, asynchronism or reject, respectively. The latter means that *s* receives a reassociation reply with flag from ap_j.
- *Verify_ANonce(*PMK_j*,s,*ap_j*,Asyn)*: The function means that AP ap_j verifies *ANonce* retrieved from STA *s* by

$$ANonce? = PRF(PMK_j, MAC_{ap_j} | MAC_s | Asyn) \quad (6)$$

STA uses Algorithm 1 to support fast reassociation with the PTKSA establishment process. If STA *s* will re-associate with the target AP ap_j, *s* checks whether *Ssyn* exists for ap_j or not. If the result is positive, *s* runs the major operation of the process. Otherwise, *s* needs to create a new *Ssyn* for ap_j. Then, *s* can run the major operation of the process. The major operation is as follows: *s* randomly generates its nonce *SNonce* and derives ap_j's nonce *ANonce* using Equation (4). Then, *s* derives PTK_j using Equation (5) and generates MIC_s using the same method in four-way handshake. After that, *s* sends a reassociation request to ap_j with *sp* and waits for a reassociation reply from ap_j. If the received reply indicates grant, *s* verifies the identity of ap_j by confirming MIC_a retrieved from the reassociation reply from ap_j. If valid, *s* sets the *Ssyn* as *Ssyn+1* and restores *Ssyn* to its storage to finish the process. Otherwise, the received reply indicates that synchronization parameters are asynchronism, *s* sets the *Ssyn* as *Asyn*, which is retrieved from the reassociation reply from ap_j. Moreover, *s* returns to the beginning of major operation of the process and repeats this major operation until obtaining a grant from ap_j.

AP uses Algorithm 2 to support the fast reassociation with the PTKSA establishment process. Upon receiving the reassociation request from STA *s*, the target AP ap_j retrieves *Asyn* for s from its storage if *Asyn* has existed. Otherwise, ap_j creates an *Asyn* for *s* and sets its initial value as 1. After that, ap_j confirms whether the value of *Asyn* and *Ssyn* are the same or not. If the result is negative, it means that the *Asyn* on ap_j and *Ssyn* on *s* are not synchronous. The ap_j replies a reassociation reply to *s* for indicating the need of

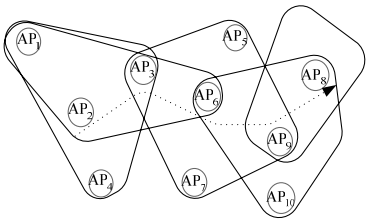

Figure 5. PMKSA distribution bases on the neighbor graph.

synchronization. Otherwise, ap_j needs to verify the identity of *ANonce* using Equation (6). If the result of Equation (6) is equal, it means that the *ANonce* which is generated by *s* is fresh and synchronous with that of ap_j. Next, ap_j derives PTK_j using Equation (5) and then verifies the identity of MIC_s retrieved from *s*. After that, ap_j generates MIC_a using the same method in four-way handshake, and then sends a reassociation reply to indicate grant. If the result of Equation (6) is unequal, ap_j replies a reassociation reply to *s* for indicating rejection.

4: Security Analysis

In this section, we exam the security analysis on the accelerated handoff mechanism.

S1: *PMK is independent*: Although APs can cache PMKSA, each AP uses a distinct PMK among other APs. Each PMK is derived using Equation (3) which is based on *CRK* and MAC addresses of STA and AP. Therefore, each *PMK* is different for the different pair of STA and AP. It brings about a better security property: if one *PMK* at one AP for one STA has been compromised, other PMKs are not affected.

S2: *Freshness, synchronization and verification of ANonce*: In the self nonce generation scheme of the fast reassociation with the PTKSA establishment process, synchronization parameters *Ssyn* and *Asyn* are critical factors to synchronize *ANonce*. If *Ssyn* retrieved from STA is the same as *Asyn*, the visited AP can use Equation (6) to confirm the validation of *Ssyn*. If the fast reassociation with the PTKSA establishment process successes, *Ssyn* and *Asyn* increase one to ensure *ANonce* in the succeeding reassociation is fresh. If *Ssyn* and *Asyn* are not the same, AP requests STA resetting *Ssyn* as *Asyn* and resends the reassociation request to ensure that the *Ssyn* and *Asyn* are synchronization.

ANonce is derived using Equation (4), where PMK and synchronization parameters *Ssyn* or *Asyn* are involved. Even though an adversary can learn all parameters in Equation (4) but the corresponding PMK, the adversary can not derive an *ANonce* by itself. Therefore, since the corresponding PMK is only kept by STA and AP, AP can confirm the validation of *ANonce* using Equation (6).

S3: *Freshness and synchronization of PTK*: The purpose of fast reassociation with the PTKSA establishment process is to guarantee the freshness and synchronize PTK. Here, we examine the property of freshness and synchronization of PTK. PTK is derived using Equation (5) such that the freshness of PTK depends on *SNonce* and *ANonce*. Since *SNonce* is generated randomly and *ANonce* is demonstrated its freshness in S2 for each process, PTK can be ensured being distinct from each fast reassociation process. Even though an adversary fixes *SNonce*, an adversary can not derive PTK due to that the value of *ANonce* can not be controlled by itself and without the knowledge of the corresponding PMK.

The purposes of *MIC* can be used to confirm the integrity of exchanged messages during the reassociation process. Here, the other purpose of MIC is to confirm that the use of PTK is the same. STA or AP must use the same PTK to derive and confirm MIC using the same method of four-way handshake, otherwise, STA or AP will take MIC as invalid.

5: PERFORMANCE ANALYSIS

In this section, we analyze the number of PMKSA distribution between Mishra et al.'s scheme and the proposed scheme. Furthermore, we evaluate the performance of the IEEE 802.11 handoff process.

5.1: THE FREQUENCY OF PMKSA DISTRIBUTION

We present the maximum and minimum number of PMKSA distribution under the condition in which STA has carried out *n* BSS transitions.

Theorem 5.1 When STA has carried out *n* BSS transitions, *n* PMKSA distributions are needed in Mishra et al.'s scheme.

Proof. In Mishra et al's scheme, after each handoff process completes, the visited AP must distribute PMKSA information about STA to the neighboring APs. In other words, if one STA has carried out *n* BSS transition, the *n* PMKSA distributions to the neighboring APs also must be invited. As Figure 5 depicted, when the visited AP of STA are in the order of AP_2, AP_3, AP_6, AP_9 and AP_8, each AP on the movement must do PMKSA distribution to its neighbors. □

Theorem 5.2 When STA has carried out *n* BSS transitions, the maximum number of PMKSA distribution *Max*(*PD*) is

$$Max(PD) = \begin{cases} \left\lceil \frac{n}{2} \right\rceil & \text{if n is odd} \\ \frac{n}{2}+1 & \text{if n is even} \end{cases} \quad (7)$$

and the minimum of PMKSA distribution *Min*(*PD*) is

$$Min(PD)=1 \quad (8)$$

Proof. The worst case occurs when STA moves from a center AP of the cluster to the second AP and then to the third AP that is the member of the selected cluster, which is depicted in Figure 4. The visited APs of STA are in the order of AP_2, AP_3, AP_6, AP_9 and AP_8 is such case. Black circular points in Figure 4 are the center AP of each cluster and gray circular points denote the members of a cluster. As Section 3.1 presented, only the center AP of a cluster needs to do PMKSA distribution

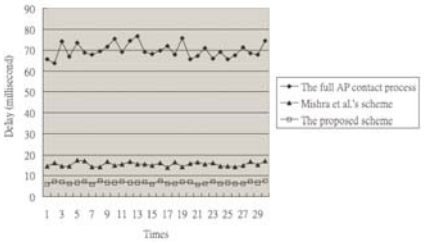

Figure 6. Handoff delay, excluding probe process.

to cluster members. Therefore, obviously, a gray point exists between two black points. We can find there are $\left\lceil \frac{n}{2} \right\rceil$ black circular points among *n* points, where *n* is odd, and there are $\frac{n}{2}+1$ black points among *n* points, where *n* is even.

The best case is when STA persists staying within the current selected cluster, i.e., STA moves to APs that are members of the selected cluster. Obviously, only one PMKSA distribution must be done by the center point of the selected cluster. □

Based on Theorem 5.1 and Theorem 5.2, we can observe that the proposed scheme has less number of PMKSA distributions comparing with that of Mishra et al.'s scheme.

5.2: THE DELAY OF THE IEEE 802.11 HANDOFF PROCESS

To evolute IEEE 802.11 handoff process, we simulate the WLAN environment using the C programming language and OPENSSL toolkit. The simulation runs on Intel P4-3.2 GHz CPU, 512 DDR SDRAM, and the Windows XP operation system. The parameters of simulation are set as follows: (1) the service radiuses of WLAN (IEEE 802.11b) is 150m and (2) the data rate of WLAN is 11Mbps.

The comparison of handoff delay among the full AP contact process, Mishra et al.'s scheme and the proposed scheme is depicted in Figure 6. In Figure 6, the numbers of simulated handoff processes are 30, where the simulated handoff process excludes probe process. Obviously, the proposed scheme needs less handoff delay than that of the full AP contact process and Mishra et al.'s scheme.

6: CONCLUSION

To accelerate handoff process, we have proposed the accelerated handoff mechanism. In the accelerated handoff mechanism, two methods are involved: (1) PMKSA distribution based on dynamic cluster selection and transition and (2) fast reassociation with the PTKSA establishment process. Overall, since the EAP-TLS authentication is eliminated using PMKSA distribution based on dynamic cluster selection and transition. Section 5 has demonstrated that PMKSA distribution using the proposed method can reduce the frequency of PMKSA distribution as comparing with Mishra et al.'s scheme using on the neighbor graph. Moreover, Section 5 has demonstrated that the proposed accelerated handoff mechanism needs less handoff delay than that of the full AP contact process and Mishra et al.'s scheme.

REFERENCES

[1] B. Aboba and D. Simon. PPP EAP TLS Authentication Protocol IETF RFC 2716, October 1999.

[2] ANSI/IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 1999.

[3] J. Edney and W. A. Arbaugh. Real 802.11 Security: Wi-Fi Protected Access and 802.11i. Addison Wesley, 2003.

[4] IEEE Standard 802.1f. IEEE. Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation. July 2003.

[5] IEEE Std 802.11i. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements. 2004.

[6] J. F. Kurose and K. W. Ross. Computer Networking: A Top-Down Approach Featuring the Internet. Addison Wesley, 2004.

[7] A. Mishra, M. Shin, and W. A. Arbaugh. Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network. Proceedings of the IEEE INFOCOM Conference, vol. 1, pp. 351-361, 2004.

[8] A. Mishra, M. H. Shin, N. L. Petroni, T. C. C. Jr., and W. A. Arbaugh. Pro-active Key Distribution using Neighbor Graphs. IEEE Wireless Communications, vol. 11, pp. 26--36, 2004.

[9] S. Pack and Y. Choi. Fast Handoff Scheme based on Mobility Prediction in Public Wireless LAN Systems. IEE Proceedings of Communications, vol. 151, no. 5, pp. 489-495, 2004.

[10] I. Ramani and S. Savage. SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks. Proceedings of the IEEE INFOCOM Conference, vol. 1, pp. 675-684, 2005.

[11] M. Shin, A. Mishra, and W. A. Arbaugh. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. Computer Communication Review, vol. 33, no. 2, pp. 93-102, 2003.

[12] M. Shin, A. Mishra, and W. A. Arbaugh. Improving the Latency of 802.11 Hand-offs Using Neighbor Graphs. Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, pp. 70-83, 2004.