

SAP: Seamless Authentication Protocol for Vertical Handoff in Heterogeneous Wireless Networks

Scott C.-H. Huang
City University of Hong Kong
Email: shuang@cityu.edu.hk

Hao Zhu
Florida International University
Email: hao.zhu@fiu.edu

Wensheng Zhang
Iowa State University
Email: wzhang@cs.iastate.edu

Abstract—802.11 standards support high data rates for a low price and thus provides an economical way for WLANs. However, 3G standards offer a much wider area of coverage that enables ubiquitous connectivity. The integration of them takes advantages from both sides and offers the possibility of achieving anywhere, anytime cost-efficient Internet access. To facilitate such integration, seamless vertical handoff is one of the major challenges because it needs to make physical movement transparent to mobile users and preserves application-level connectivity. Previous works did not consider the impact of authentication mechanisms on the performance of vertical handoff, especially on its delay. In a 3G-WLAN integration environment, since 3G and WLAN may use different authentication servers, when a mobile station hands over across them, certain authentication procedure needs to be performed. According to the literature, such authentication delay may be as high as hundreds of milliseconds, which is intolerable for delay-sensitive applications. We present SAP, the Seamless Authentication Protocol for vertical handoffs in wireless heterogeneous networks, to reduce this delay. Simulation results show that SAP significantly reduces the delay caused by authentication procedures in vertical handoff.

I. INTRODUCTION

Technological development of wireless networks have brought a deep change in our lifestyle. In addition to traditional voice services, many data services (e.g. WWW, IP multimedia) have been carried over the wireless terrestrial networks as a last-mile access of today's Internet. The wide deployment of wireless infrastructures facilitates the accessibility to the IP-based data and therefore moves one step forward toward making it available anywhere anytime.

The 802.11 standards allow the realization of economic Wireless LANs that support data rates anywhere from 1Mbps to 54 Mbps based on the distance to the access point. However, 802.11 access

points can cover areas of only a few thousand square meters, making them suitable for enterprise networks and public hot-spots such as hotels, coffee shops, and airports. On the contrary, wireless cellular networks, built using the 3G standards, offer a much wider area of coverage that enables ubiquitous connectivity. However, 3G cellular networks require significant capital investments and support limited peak rates that range from 64 Kbps to nearly 2Mbps. The two technologies offer characteristics that complement each other perfectly. Thus, the combination of 3G and WLAN technologies offers the possibility of achieving anywhere, anytime cost-efficient Internet access, bringing benefits to both end users and service providers.

In such heterogeneous wireless systems, one of the major challenges is seamless vertical handoff. Seamless handoff is involved in the availability of the mobile terminal to successively attach to different access points or base stations in wireless heterogeneous networks. Such procedures need to make the physical movement transparent to mobile users and preserve application-level connectivity. In order to achieve seamless handoff, several issues such as handoff metrics, handoff decision algorithms, and mobility management need to be addressed.

There have been several works on vertical handoff in the literature. In [20], a roaming scheme that considered the relative bandwidth of WLAN and GPRS was proposed. In [17] a detailed vertical handoff signaling procedure was presented. [24] proposed a mobility management system that integrates a connection manager to maintain a connection without additional network infrastructure support. [14] provided a quantitative analysis of a mobile IPv4-based WLAN-GPRS handoff prototype

and identified a number of side effects related to the link layer and routing mechanisms. They also presented the impact of handoff on UDP and TCP data traffic as well as on mobile IP signaling itself. However, these works did not consider the impact of authentication mechanisms on the performance of vertical handoff, especially on its delay. Wherever a vertical handoff takes place, for the purpose of security, an authentication process should be performed to ensure the identity of the mobile station. Since different access networks may use different authentication servers and protocols, when a mobile station hands over from one access network to another (e.g. from 3G to WLAN), certain authentication procedure (e.g. 802.11i [6]) needs to be performed. As shown in [9], the delay of such procedure may go up to hundreds of milliseconds, which is intolerable for delay-sensitive applications such as VoIP or streaming applications.

In this paper, we focus on a fast authentication mechanism in vertical handoff, as how to speed up this process to support realtime/multimedia applications in wireless heterogenous networks is a fundamental issue. The basic ideas are as follows. Without loss of autonomy, we have the two different authentication servers share a common secret, which is to be used later to generate a temporary handoff key. During a vertical handoff (e.g. the mobile station switches from 3G to WLAN), the access point uses this temporary handoff key obtained from its authentication server to admit the mobile station for a short while (e.g. up to 10 seconds). Meanwhile, the full authentication is being performed simultaneously. Based on the result of the full authentication, the access point decides whether or not to permanently admit the mobile station. Since authentication servers take no part in temporary authentication, the delay of temporary authentication is significantly less than that of full authentication. As a result, the impact of full authentication on vertical handoff delay is greatly mitigated. On the other hand, since full authentication is being performed on the background, the security loophole of temporary authentication is quite limited. Following these ideas, we design three fast authentication protocols, SAP v1,v2,v3, and their corresponding key management schemes. In particular, SAP v1 is the most secure while v2 has

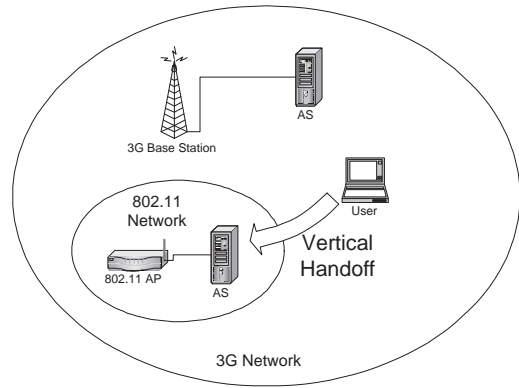


Fig. 1. System Model

the least key management overhead. SAP v3 is designed to balance the advantage and disadvantage of SAP v1 and v2. SAP v3 forms a complete spectrum of tradeoffs between security and efficiency. We also provide a detailed security analysis of the proposed protocols and compare their performance through extensive simulations. Simulation results show that, compared with full authentication, SAP significantly reduces authentication delay in vertical handoff.

The rest of this paper is organized as follows: Section II is regarding the background and motivation, on which our protocols are based. In Section III we present our protocols, namely the full authentication and SAP v1,v2,v3, in vertical handoff. In Section IV we do security analysis. The results of performance evaluation are shown in Section V, Section VI lists related work, and Section VII concludes our work.

II. BACKGROUND AND MOTIVATION

A. Background

In this section, we start with introducing the system model of our core scenario. Then we review the handoff procedures in both 3G and 802.11i networks with particular emphasis on security considerations.

1) *System Model*: As explained in [12], we consider the loosely-coupled interworking since its complexity and cost of deployment are lower than the tightly-coupled one. Specifically, as shown in Figure 1, the gateway connects to the Internet and there is no direct link from 802.11 elements (e.g. Access Points) to 3G network ones (e.g. Packet Data Serving Nodes or 3G core network switches). From a security point of view, this means different networks connect different authentication servers

(ASs) through the Internet. The users that access services of the 802.11 gateway include users having locally signed on or mobile users visiting from other networks. This model is quite light-weight, flexible, and practical. A vertical handoff takes place when a mobile station switches between 802.11 and 3G networks.

2) *Authentication in 3G*: 3G authentication lies in the *Routing Area Update* process. Take *Intra-SGSN (Serving GPRS Support Node) routing area update* for example, the mobile station sends the routing area update request to the target RNC(Radio Network Controller), and the RNC forwards it to the target SGSN. The target SGSN then needs to authenticate the mobile station to determine whether or not the request can be accepted. *Inter-SGSN routing area update* is no exception either. Authentication needs to be performed when the routing area update request reaches the target SGSN.

The network access security specified in 3GPP has three building blocks: *authentication and key agreement (AKA)*, *UMTS encryption algorithm (UEA)*, and *UMTS integrity algorithm (UIA)*. We only focus on the authentication process of AKA. This process provides mutual authentication for both the users and the network. Two keys are generated in 3GPP AKA: CK for encryption and IK for integrity. There is a secret key K , shared by the user and the network and available only to the *authentication center (AuC)* in the user's *home environment (HE)* and the USIM (Universal mobile telecommunication system Subscriber Identity Module) on the mobile station.

Upon receiving an authentication request from a visitor location register(VLR) or SGSN, HE/HLR distributes to SGSN/VLR a set of authentication vectors (AVs), ordered based on the sequence number. Each AV is good for one AKA between VLR/SGSN and the USIM. To authenticate a user, VLR/SGSN retrieves the next available AV. Based on the secret material in the AV, the mobile station can authenticate the network. Then, the mobile station generates the response and sends it back. VLR/SGSN then authenticates the user by comparing the received response with the expected one.

3) *Authentication in 802.11 WLANs*: An AP and associated STAs form a *basic service set (BSS)*. A collection of APs connected by a wired network can

extend a BSS into an *extended service set (ESS)*. If a STA wants to join an ESS network, it must be authenticated by showing its credentials to AP. AP then passes these credentials to a fixed *authentication server (AS)* to verify these credentials. Upon receipt of AS's decision, AP will either associate or reject the aspirant STA.

A typical authentication procedure in 802.11i usually involves EAP/TLS authentication and RADIUS back-end protocol. In fact, concrete EAP authentication methods and back-end protocols are beyond the scope of 802.11i, but EAP/TLS is the *de facto* authentication protocol and RADIUS is the *de facto* back-end transport protocol for EAP over IP networks. There will be certain latency involved in the whole handoff process, which can be divided into two phases:

Probe phase: The mobile station (STA) scans through all possible channels to find access points (APs) of good signal strength. It is called the passive scan. These beacon messages are usually sent periodically by APs at a rate of 10ms. Also, STA can actively send probe requests in the first place to get responses from APs. It is called active scan. The delay caused by scanning is called the *probe delay*, which is sometimes of magnitude 100ms. In our system, by letting 3G BSs share the channel assignment of each 802.11 AP, the probe delay can be significantly reduced.

Reassociation phase: After finding the preferred AP, STA tries to associate with it and performs *context transfer*. The new AP will first try to contact the old AP and get the *security context* (such as encryption key, ...etc). This can be done only if there is a trust relation between these two APs. Specifically, if the two APs are within the same ESS, then the old AP can pass the entire security context to the new AP to reduce the authentication latency. Otherwise, it means the two APs have different ASs associated with them. The new AP should then ask for its AS to authenticate STA, and performs a full authentication. Considering the transmission delay between AP and AS plus the processing delay at AS, the authentication delay is usually of magnitude of several hundred milliseconds or even several seconds [9].

B. Motivation

Since 3G networks and WLANs are loosely integrated through the Internet and they may not share the same AS, this heterogeneity makes context caching [19] impractical. For instance, existing protocols of this type such as IAPP [4], [5] or Seamoby [8] all assumed the homogeneity of the networks to perform Layer 2 context (e.g. MAC address of the old and new APs, encryption keys) transfer during a handoff. Other works such as Bargh *et al* [10] assumed the homogeneity of service-providing networks to perform proactive key distribution, too. In the example of Figure 1, none of these protocols work due to the lack of a shared AS (Note that different ASs have different security contexts). Therefore, when a mobile station roams (1) from 3G network to WLAN, especially when the WLAN is not covered by any 3G network¹, (2) from WLAN to 3G, or (3) between WLANs using different ASs, full authentication cannot be avoided and the resulting delay becomes the bottleneck of the handoff process. For example, the typical latency of a full authentication in WLAN can be as high as 800ms (Arbaugh [9]), which is much greater than the maximal tolerable latency for delay-sensitive applications (e.g. 50ms for VoIP). SAP was mainly designed for fast authentication in such vertical handoff. Our goal is to reduce the full authentication delay in order to decrease the impact of vertical handoff on delay-sensitive applications.

III. THE PROPOSED PROTOCOLS

For brevity and clarity, we mainly describe the protocols as the solution for vertical handoffs from 3G to WLAN. The solution can be easily extended and applied to other types of handoff between access networks associating with different ASs.

A. Sharing Secrets Between ASs

To facilitate fast authentication in vertical handoff, one approach is to have the new AP/BS utilize the security context of the mobile station from the old AP/BS. To preserve autonomy, we have different authentication servers share a ‘‘SAP master key’’ SAP_M instead of the whole credential database. This key is updated periodically with a relatively large interval (such as every 30 minutes),

¹This can happen on the boundary of a 3G cell or in concrete buildings where 3G’s signal strength is very weak.

and then ASs distribute it to all of their APs/BSs. Note that this technique is practical because SAP_M is independent of mobile stations and no context transfer is involved. Upon receipt of new SAP_M , each AP computes the SAP temporary key sap_i and distributes it to every mobile station it is currently associated with. sap_i is periodically updated by AP and then distributed to the mobile stations. This update should be frequent enough (e.g. every several minutes) to discourage attackers and provide better security.

B. Notation and Key Hierarchy

We list the notations and key hierarchy here. The cryptographic keys involved are shown in Table I. We use *pseudo-random functions* [15] to generate cryptographic keys. We use the following notation $h_i(X)$ to represent $F_X(i)$ for $0 \leq i \leq 6$, where F is a pseudo-random function constructed using techniques in [15]. For example, $h_3(X)$ means $F_X(3)$. The key hierarchies are shown in Figure 3, 2, and 4. As shown in Figure 3 for instance, the pairwise master key PMK is computed by feeding N_{AS}^1 and MK to the pseudo-random function h_0 . Similarly, the key MK' is obtained by feeding MK and N_{STA}^1 to h_0 .

abbreviation	full name & description
MK	master key, shared by STA and AS only
PMK	pairwise master key, shared by STA,AP,AS
SAP_M	SAP master key, shared by AS & AP
sap_i	SAP temporary key, shared by STA & AP
K_{STA}	SAP session key, shared by STA and AP
N_X^i	The i-th nonce generated by entity X
SAK	subgroup assignment key, shared by ASs and APs
k_0, k_1, \dots	subgroup keys, shared by ASs and APs

TABLE I
NOTATION

C. Full Authentication

Here we present our full authentication scheme as a building block for all of our schemes. The full authentication scheme authenticates both the mobile station and the access point and establish security context between them (in this case it is the pairwise master key PMK).

1) *Full authentication*: Full authentication is a 4-way handshake protocol, whose goal is to provide mutual authentication and establish the security context (PMK) between STA and AP.

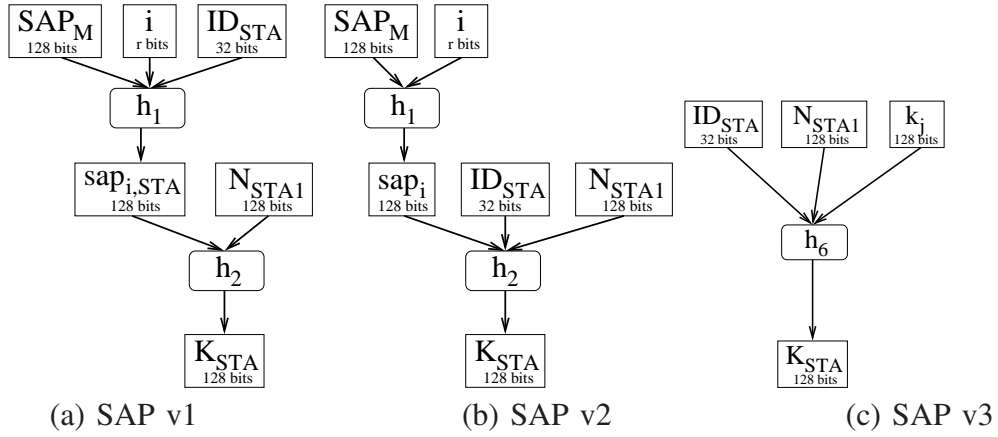


Fig. 2. The key hierarchy of SAP v1, v2 and v3

D. SAP v1

The basic ideas behind the SAP authentication are: (1) using temporary handoff keys to make the reassociation process seamless (2) performing a full authentication simultaneously in order to provide sound security.

This temporary handoff key needs to be carefully maintained to deter possible attacks and ensure good security during the temporary reassociation period. SAP v1 can be divided into the following two parts: (1) key management at both AS and AP (2) SAP authentication. The details are as follows.

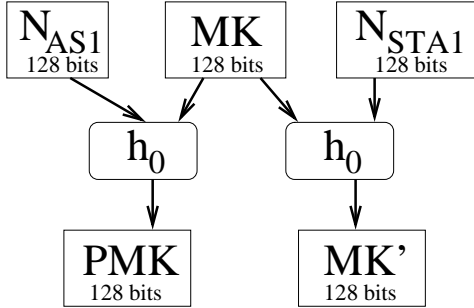


Fig. 3. Key hierarchy of full authentication

1) *SAP v1 key management*: The key management is vital in providing security for SAP authentication, which will be described later. We present the scheme for the authentication server (AS) and access point (AP) respectively as follows.

a) *Key management at AS*:

- AS generates SAP_M . This new SAP_M should be totally irrelevant to old ones.
- AP \leftarrow AS: SAP_M

These two steps need to be done periodically by AS (e.g. every 30 min).

b) *Key management at AP*: For each STA, AP does the following:

- AP computes $sap_{i,STA} = h_1(SAP_M || i || ID_{STA})$ locally.
- STA \leftarrow AP: $sap_{i,STA}$

For each mobile station STA, AP periodically computes the key $sap_{i,STA}$ and unicasts it to STA. These two steps need to be done periodically by AP with a relatively smaller period (e.g. every 2 min). Once an STA gets associated to some AP, it gets the temporary sap key $sap_{i,STA}$ at the same time. From now on, no matter where it goes, this key can be used to perform temporary association when it roams across different service providers. There is no need to use key management scheme of any kind to have all APs share this key, as every AP can simply compute it locally. However, STA cannot compute it locally as it does not know SAP_M . Thus, this key must be given by some AP it is associated with before handoff. The overhead comes from the periodical update of SAP_M among ASs and the unicast of $sap_{i,STA}$ by its associated AP. The former overhead is nearly negligible since it does not have to be done frequently (e.g. once every 5 mins is enough in most cases) and it involves broadcast only. The latter has relatively more overhead due to the unicast compared with other SAP versions, but every AP can only be associated with limited STAs and the benefit it brings greatly exceeds the overhead it increases.

2) *SAP v1 authentication*: SAP v1 authentication is designed for fast handoff between heterogeneous networks. The core of this scheme is to use a temporary handoff key $sap_{i,STA}$ to establish a temporary association while the full authentication is being done at the same time. Note that this temporary association must provide certain security, and using open association without any authentication in this case is definitely infeasible. The reason for that is attacker may repeatedly launch the attack of temporary association request if it is not secured. This may exhaust system resources and disrupt the whole network.

SAP v1 Authentication Protocol

1. STA generates N_{STA}^1 and computes $K_{STA} = h_2(sap_{i,STA} || N_{STA}^1)$
2. STA \rightarrow AP: ID_{STA}, N_{STA}^1 , “SAP Authentication Request”, $HMAC_{K_{STA}}(ID_{STA})$
3. AP computes $K_{STA} = h_2(h_1(SAP_M || i) || ID_{STA}) || N_{STA}^1$ and verifies HMAC. If it succeeds, AP temporarily associates with STA using K_{STA} .
4. STA \leftarrow AP: N_{AP}^1 , “SAP Authentication Success”, $HMAC_{K_{STA}}(N_{AP}^1 || ID_{STA})$
5. STA verifies HMAC and both STA and AP change association status to SAP association.
6. Perform full authentication procedures.
7. If full authentication succeeds, STA and AP change association status to full association and use full authentication key PMK for encryption henceforth. If it fails, AP de-associates with STA after SAP association expires.

This quick 2-way SAP authentication (steps 1-4) temporarily associates STA with the new AP. In the meantime, AP also tunnels ID_{STA}, N_{STA}^1 to AS to perform full authentication (step 5). We only allow SAP association up to a predefined time (e.g. 10sec). After that, if full authentication fails, AP will simply de-associate with STA (step 6).

E. SAP v2

SAP v2 key management:

a) Key management at AS:

- AS generates SAP_M . This new SAP_M should be totally irrelevant to old ones.
- AP \leftarrow AS: SAP_M

These two steps need to be done periodically by AS (e.g. every 30 min).

b) *Key management at AP*: AP does the following for everybody:

- AP computes $sap_i = h_1(SAP_M || i)$ locally.
- STA \leftarrow AP: sap_i

Different from SAP v1, now AP basically computes sap_i and broadcast to every STA it is currently associated with. The period should be relatively small as well (same as SAP v1). This option is more efficient because it uses broadcast instead of unicast to maintain the key sap_i , but it is less secure than SAP v1.

F. SAP v3: r -SAP

SAP v3 is a tradeoff between v1 and v2. v1 being the most secure and v2 being the most efficient, SAP v3 tried to balance security with efficiency. SAP v3 has a parameter r representing its randomness. For this reason, we also denote it by r -SAP.

1) *r -SAP key management*: AS generates a *Subgroup Assignment Key* SAK and applies an r -bit pseudo-random function h_3 to group STAs into 2^r subgroups (h_3 takes an arbitrary length input and outputs r bits). Each mobile station is given a subgroup ID j when it first gets associated with an AP. The subgroup ID, j , will be generated by AP: $j = h_3(ID_{STA} || SAK)$, where h_3 is an r -bit pseudo-random function. In addition to SAP_M and SAK , AS also generates subgroup keys k_0, k_1, \dots for each subgroup (each subgroup ID j corresponds to one k_j). Note that in an r -SAP scheme there will be 2^r subgroups (therefore there are 2^r subgroup keys).

SAK is shared only by ASs and APs, so no STAs have any knowledge about it. The k_j 's are shared by ASs and AP as well, and each STA only knows the k_j of **its own** j . The STAs do not have any knowledge about other k_j 's. Moreover, they will be updated periodically with SAP_M for security reasons.

a) Key management at AS:

- AS generates SAP_M, SAK , and $\{k_j | 0 \leq j < 2^r\}$. (j is the subgroup ID)

- AP \leftarrow AS: $SAP_M, SAK, \{k_j\}$

These two steps need to be done periodically by AS (e.g. every 30 min).

b) *Key management at AP and STA*: AP uses a pseudo-random function h_4 to generate n_j 's (for each subgroup). h_4 outputs bit strings of the same length as k_j 's. h_4 is shared by APs and ASs only (STAs have no knowledge about it). h_5 is another *PRF* used by STAs to update k_j 's. h_5 is known to everybody. The key management procedures are as follows:

- AP computes $n_j = h_4(SAP_M || j || k_j)$ for all j , for all $0 \leq j < 2^r$
- STA \leftarrow AP: $\{E_{k_j}(n_j) | 0 \leq j < 2^r\}$
- AP updates $\{k_j\}$ by $k_j \leftarrow h_5(k_j \oplus n_j)$, for all $0 \leq j < 2^r$
- each STA decrypts his own piece of message, gets his own n_j , and updates his own k_j by $k_j \leftarrow h_5(k_j \oplus n_j)$

With respect to each subgroup, AP broadcasts the encrypted message to every STA it is currently associated with. Since each STA only possesses its own subgroup key, it can only decrypt his own piece of message.

2) *r-SAP authentication*:

SAP v3 Authentication Protocol

1. STA \rightarrow AP: $ID_{STA}, N_{STA}^1, HMAC_{k_j}(ID_{STA} || N_{STA}^1)$, "SAP Authentication Request"
2. AP computes its subgroup ID (by plugging in $h_3(ID_{STA} || SAK)$) and locates its corresponding k_i . AP then uses this k_i to verify HMAC. If it succeeds, AP computes $K_{STA} = h_6(ID_{STA}, N_{STA}^1, k_j)$ and temporarily associates with STA using K_{STA} .
3. STA \leftarrow AP: $N_{AP}^1, HMAC_{K_{STA}}(N_{AP}^1 || ID_{STA})$, "SAP Authentication Success"
4. STA verifies HMAC and both STA and AP change association status to SAP association.
5. Perform full authentication procedures.
6. If full authentication succeeds, STA and AP change association status to full association and use full authentication key *PMK* for encryption henceforth. If it fails, AP de-associates with STA after SAP association expires.

This quick 2-way SAP authentication (steps 1-5) temporarily associates STA with the new AP. In the meantime, AP also tunnels ID_{STA}, N_{STA}^1 to AS to perform full authentication (step 6). We only allow SAP association up to a predefined time (e.g. 10sec). After that, if full authentication fails, AP will simply de-associate with STA (step 7).

IV. SECURITY ANALYSIS

A. Full Authentication

We use the following four points described in [13] to analyze our scheme:

- 1) Robust method of proving identity that cannot be spoofed
- 2) Method of preserving identity over subsequent transactions that cannot be transferred
- 3) Mutual authentication
- 4) Authentication keys independent of encryption ones

(Rule 1) AS proves its authenticity to STA at step 2 as it sends the encrypted message $E_{MK}(N_{STA}^1, N_{AS}^1)$ using the master key *MK*,

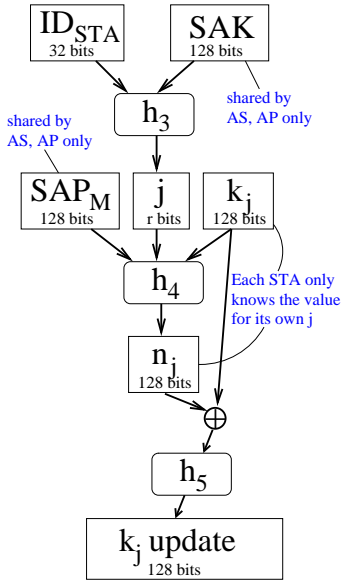


Fig. 4. SAP v3 k_j update

which is assumed to be shared by STA and AS only. STA proves its identity to AS at step 4 by showing its capability of generating the pairwise master key PMK , which is generated from MK , too. (Rule 2) After finishing the full authentication, the security context PMK will be established and subsequent transactions will be encrypted with its derived key. (Rule 3) Mutual authentication is preserved because both parties shows its knowledge of MK , as described in the analysis of rule 1. Either parties lack of this knowledge will cause the full authentication scheme to fail. (Rule 4) The authentication key in our scheme is actually MK , and the encryption key is some other key derived from PMK . Thus, the authentication and encryption keys are independent of each other because PMK is also derived from N_{AS}^1 , which has never been disclosed throughout our scheme. Besides the basic requirements, we also analyze the security and robustness of our scheme by considering the following three types of attacks: impersonating, eavesdropping, and replay attacks. The details will be presented as follows.

1) *Impersonating attack*: The adversary may impersonate either party to obtain cryptographic data and steal sensitive information. The full authentication scheme is robust against such attack for the following reasons:

(1) If the attacker impersonates STA, then the data

it needs to forge will be the 3-tuple

$$\{ID_{STA}, N_{STA}^2, HMAC_{PMK}(ID_{STA} || N_{AS}^1 || N_{STA}^2)\}$$

Within the data, ID_{STA} is open to public and N_{STA}^2 can be generated, but the HMAC cannot be forged because PMK and N_{AS}^1 are both unknown. Same HMAC has never been sent before, so there is no way to replay old messages either.

(2) On the other hand, the attacker cannot impersonates AP, as $E_{MK}(N_{STA}^1, N_{AS}^1)$ cannot be generated due to attacker's lack of knowledge about MK . Therefore, STA will not be able to decrypt the message and correctly match it with N_{STA}^1 .

2) *Cryptanalysis attack*: The adversary can passively listen to all the communication and try to perform cryptanalysis and get the cryptographic keys. However, for the same reason as in the impersonating attacks, the nonce N_{AS}^1 will never be shown and the data the attacker can collect is not even enough for performing the *known plaintext attack* [22]. Thus our scheme is secure against eavesdropping.

The adversary can also actively send test messages to collect useful data for cryptanalysis, but we claim that such attempt is futile because the data an attacker can collect are not enough to perform the *chosen plaintext attack* [22]. An attacker can use some existing ID_{STA} , generate a nonce N_{STA}^1 , ask for full authentication, and get some sample data. However, the data the attacker can get, namely $E_{MK}(N_{STA}^1, N_{AS}^1)$, are totally useless because the nonce N_{AS}^1 has never appeared before and the encrypted message along with ID_{STA}, N_{STA}^1 are enough for nothing more than the *ciphertext only attack* [22].

3) *Replay attack*: If the adversary has a wireless sniffer that is able to capture all the frames sent between an access point and a mobile station, then it can launch the replay attack, trying to re-send an old message that has been authenticated without knowing the contents. Our full authentication scheme is robust against such attacks because of the use of nonces. The nonce N_{AS}^1 is generated by the authentication server and the attack has no control over it at all. In a replay attack, a new N_{AS}^1 will be generated and it will be completely independent of the old ones. Without the knowledge of it, the attack can by no means be authenticated unless it is able to get MK and decrypt the message. For this we

claim that the full authentication scheme is robust against replay attacks.

B. SAP Key Management and Authentication

We examine the three attacks described earlier.

1) Impersonating Attack:

a) *SAP v1*: If the attacker impersonates STA, then it has to forge the following data:

$$ID_{STA}, N_{STA}^1, HMAC_{K_{STA}}(ID_{STA})$$

However, the *HMAC* key K_{STA} cannot be faked because it equals $h_2(sap_{i,STA} || N_{STA}^1)$ and the attacker has no knowledge about $sap_{i,STA}$. On the other hand, if the attacker impersonates AP, it has to send the following data:

$$N_{AP}^1, HMAC_{K_{STA}}(N_{AP}^1 || ID_{STA})$$

This cannot be faked, as the attacker cannot generate a nonce ID_{STA} along with its corresponding *HMAC* without knowing the MAC key K_{STA} . Therefore, *SAP v1* is robust against the impersonating attack.

b) *SAP v2*: The strength of *SAP v1* is originated from the attacker's lack of knowledge about $sap_{i,STA}$, but this comes at a price. AP has to unicast $sap_{i,STA}$ to every station individually on a regular basis. This may be a significant overhead in some scenarios. *SAP v2* trades in the security on this aspect for more efficiency by broadcasting a shared root key sap_i to every station and having them compute the SAP key K_{STA} themselves. This results in the fact that everybody is capable of computing every other node's SAP key in the same group. As a result, everybody can impersonate everybody (even the AP) as long as they are in the same group. More specifically, if an attacker wants to impersonate STA (which belongs to the same group as the attacker), all he has to do is generate N_{STA}^1 and compute K_{STA} by plugging in $K_{STA} = h_2(ID_{STA} || N_{STA}^1 || sap_i)$. Note that K_{STA} is computable because ID_{STA} is open to everybody, N_{STA}^1 can be generated randomly, and sap_i is known to the attacker due to being in the same group. Once K_{STA} is computed, *HMAC* can be computed as well and the authentication message (at step 2) can be forged completely. On top of that, even the AP of the same group can be impersonated as well since K_{STA} can be computed for the same reason. Therefore, *SAP v2* is vulnerable to the impersonating attack.

c) *SAP v3*: *SAP v1* is robust against impersonating attacks while *SAP v2* is not, but the key management cost of *SAP v1* is significantly higher than that of *v2*. In *v1*, each AP has to compute $sap_{i,STA}$ for every STA and unicast to them periodically. In *v2*, each AP only has to compute sap_i for all STAs and broadcast to them in one shot, but the tradeoff is less security. However, even though *v2* is not that secure in this case, it only lasts for a few seconds. The full authentication still needs to be completed, and, if it fails, SAP connection will be de-associated immediately, too.

SAP v3 is a further tradeoff between *v1* and *v2*. Both its security and key management cost are between *v1* and *v2*. In the *r*-SAP, the SAP root key has been broken into 2^r pieces $\{k_1, k_2, \dots\}$, so the probability that an attacker can successfully impersonate a node or an AP equals $1/2^r$ and its overhead increases correspondingly too. If h_6 is secure enough, to impersonate a node or an AP, the attacker needs to be able to compute the SAP key K_{STA} , which is derived from the root key k_i . The security of *SAP v3* thus solely depends on the knowledge of K_{STA} and the security of h_6 , which can even be kept secret to increase security (though we didn't assume that and it is not absolutely necessary.) We do not consider the case of launching a cryptanalysis attack on these hash functions because it is beyond the scope of our discussion. The probability that the attacker knows the same k_i as the victim node is $1/2^r$, so the probability of successfully impersonating a node is $1/2^r$ too.

d) *Hiding of subgroup information*: The security of *SAP v3* comes from the hiding of subgroup information. The hiding of the subgroup assignment key *SAK*, prevents every node from knowing its group ID. Also, the SAP root key k_i 's will never be exposed directly in the messages, resulting in the inability of determining whether two node belong to the same group or not. These properties are carefully designed so as to discourage attackers. If these data were not hidden, an attacker can simply listens to the communication, wait for the node that belongs to the same group, and launch the impersonating (or other) attacks. In *SAP v3*, it is not easy. If the pseudorandom function h_3 is chosen properly and *SAK* is kept secret, the attacker cannot do anything

other than guessing the grouping information. This helps discourage attackers while preserving most of the security.

2) *Cryptanalysis attack*: SAP v1 is robust against cryptanalysis attack. SAP v1's robustness is due to the use of HMAC and the secrecy of the MAC key K_{STA} . Since attacker has no knowledge about K_{STA} , the best attack he can launch is the direct *birthday attack* on the underlying hash function. However, if this underlying hash function is chosen carefully (such as MD5), the practical usefulness is negligible [11], [16]. Different from v1, SAP v2 is vulnerable to cryptanalysis attack since K_{STA} only depends on $ID_{STA}, N_{STA}^1, sap_i$, in which I_{STA} is public, N_{STA}^1 can be generated, and sap_i is known to all STAs associated with the current AP. SAP v2 is vulnerable to the known plaintext attack because the adversary can record all communication messages between an STA of the same group and get many 3-tuples $ID_{STA}, N_{STA}^1, HMAC_{K_{STA}}(ID_{STA})$ for cryptanalysis. On top of that, the adversary can even fake the communication message and record AP's response message $N_{AP}^1, HMAC_{K_{STA}}(N_{AP}^1 || ID_{STA})$ for cryptanalysis, too. However, the adversary cannot perform the chosen plaintext attack, as he has no control over the nonce N_{AP}^1 , nor can he perform the replay attack, as the nonce is used throughout the scheme to provide freshness.

SAP v3 needs to be examined and analyzed in a probabilistic fashion. Similar to earlier discussion on impersonating attack, in r -SAP, the adversary has knowledge about K_{STA} with probability $1/2^r$. Therefore, with this probability the adversary can perform the known plaintext attack. SAP v3 is robust against chosen plaintext and replay attack for the same reasons as v2.

As a result, we conclude that SAP v1 is robust against cryptanalysis, v2 is vulnerable to known plaintext attack, and v3 is vulnerable to known plaintext attack with probability $1/2^r$. Between v1, v2, and v3, the least secure case is v2. However, the worst thing that may happen is the misuse of the short temporary association, which might not cause big problems since full authentication is still being processed. SAP v1, v2, v3 altogether provide a complete spectrum of tradeoffs between security

and efficiency.

V. SIMULATION RESULTS

In this section, we evaluate the performance of the full authentication and SAP v1, v2 and v3. Our simulation is based on ns-2 [1]. The simulation scenario is similar to Figure 1. We assume there are three APs residing within the coverage area of the 3G base station. The WLAN service provider is supposed to be independent of the 3G service provider. Thus, APs and BS have different authentication servers (ASs). Mobile users are assumed to have subscribed both WLAN and 3G services. A mobile terminal switches from 3G to WLAN to enjoy a high bandwidth when it goes into the coverage area of the corresponding AP, and hands over from WLAN to 3G when it moves out of the AP's coverage range. We assume vertical handoff from 3G to WLAN follows a Poisson process, and then the handoff interval is exponentially distributed with a mean handoff interval. Before AP admits the mobile station, the identity of mobile station should be verified by AS of the WLAN service provider. We assume the service latency of each authentication request is exponentially distributed with a mean service time. Each AP connects to AS through the Internet, and the average end-to-end bandwidth and delay of the link is 10 Mbps and 10 ms respectively. For simplicity, the channel capacity of each WLAN is assumed to be 2 Mbps, and all data packets are served with 802.11 DCF [2].

We first evaluate the performance of SAP by comparing it with the full authentication protocol. The performance metric is the authentication delay, which is equal to the time interval from AP sending the authentication request to the time the mobile station being admitted by AP. Note that we do not consider the authentication delay for malicious mobile stations since the delay will be infinite. We also study the impact of mean handoff interval and mean service time. Then we evaluate the communication overhead of key distribution schemes used by SAP v1, v2 and v3.

A. The Authentication Delay

We compare the authentication delay of full authentication and SAP v1, v2 and v3. We first fix the mean service time of AS to 10 ms and evaluate the authentication delay as the function of the mean

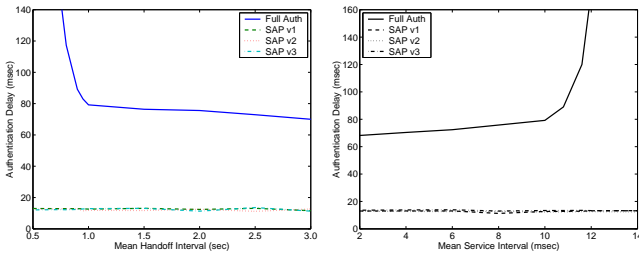


Fig. 5. The authentication delay of the full authentication and SAP v1, v2 and v3

handoff interval. We change the aggregated mean handoff interval² from 0.5 second to 3.0 seconds. As shown in Figure 5 (a), the authentication delay of full authentication is much longer than that of SAP v1, v2 and v3. Specifically, when mean handoff interval is quite small (say less than 1.0 second), the authentication delay increases a lot. This long delay mainly comes from the queuing delay at AS due to limited processing capacity. In this case, AS is the performance bottleneck. When the frequency of handoffs decreases, meaning that the mean handoff interval increases, the queuing delay at AS becomes almost zero. However, due to the service latency at AS plus the transmission and propagation delay between AP and the BS, the authentication delay of full authentication is still much longer than that of SAP v1, v2, and v3. On average, the delay of full authentication is around 80 *ms*, which is much longer than the delay requirement of most real-time applications.

According to Figure 5(a), compared to full authentication, SAP v1, v2 and v3 are not sensitive to mean handoff interval at all. The reason is that each authentication is processed by AP, and full authentication takes place on the background. Since AP can temporarily admit the mobile station according to locally stored SAP keys, the delay of background full authentication can be concealed. Since SAP v1, v2 and v3 differ only in key distribution, we can see that their delays are very close to each other.

We then evaluate the delay of full authentication and SAP v1, v2 and v3 as a function of the mean service time of AS. We fix mean handoff interval to 1.0 second. The mean service time is changed from 2 *ms* to 14 *ms*. The results are shown in Figure

²It is because there are more than one APs that share the same AS.

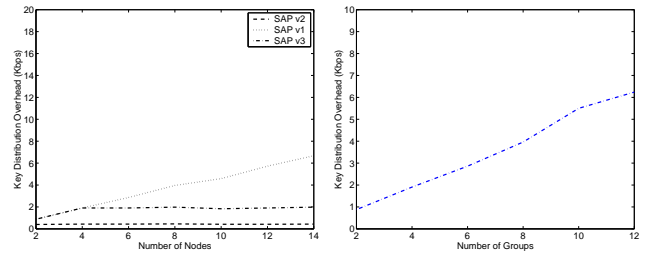


Fig. 6. The overhead of key distribution in SAP v1, v2 and v3

5 (b). For full authentication, we can see that the delay increases rapidly if the mean service time is greater than 10 *ms*. This can also be explained by the queuing delay at AS. Similar to Figure 5 (a), the delays of SAP v1, v2 and v3 are not sensitive to the processing capacity of AS. It is also because that AP does not need to contact with AS to perform temporary admission.

B. The Overhead of Key Distribution

Compared to full authentication, SAP v1, v2, and v3 require AP to periodically update the temporary handoff key, which should be distributed to each mobile station in the WLAN. In this section, we evaluate the communication overhead of key distribution in SAP v1, v2 and v3. The overhead (in bps) is equal to the total amount of traffic (in bits) used for key distributions divided by the simulation time³. The key update interval is assumed to be 2 seconds. We evaluate the overhead as a function of the number of nodes. The overhead of SAP v3 is further studied with various number of groups.

We first study the overhead under different number of nodes. For SAP v3, the number of group is assumed to be 4. As shown in Figure 6 (a), SAP v1 has the least overhead while the SAP v2 has the most. Since SAP v1 distributes the temporary handoff key to each mobile station simply using a broadcast, the overhead can be minimized. However, as stated in Section IV, SAP v1 has the least security. Since SAP v2 distributes this key to each mobile station one-by-one through unicast, the overhead increases as the number of mobile stations increases. Fortunately, since the size of each key distribution message is quite small (less than 30 bytes), the overhead of SAP v2 is not large

³Note that the amount of traffic includes MAC layer frames (e.g. RTS, CTS, ACK...[2])

provided that the number of nodes is moderate. SAP v3 balances the tradeoff between key distribution overhead and security. As shown in Figure 6 (a), as the number of nodes increases, the overhead of SAP v3 is much smaller than that of SAP v2. Since the overhead of SAP v3 is directly related to the number of groups, it does not increase if the number of nodes is more than the number of groups. We also evaluate the overhead of SAP v3 as a function of number of groups. As shown in Figure 6 (b), the overhead of SAP v3 almost linearly increases as the number of groups increases. Thus, the communication overhead of SAP v3 can be well controlled by adjusting the number of groups.

VI. RELATED WORK

In 802.11 Wireless LANs, 802.11i [6] defined the *Robust Security Network* (RSN), The authentication scheme of 802.11i was based on IEEE 802.1X [3], which employed *Extensible Authentication Protocol* (EAP), allowing different authentication mechanisms to establish layer 2 session key dynamically. IEEE 802.1X does not define the way that EAP messages are passed between the authenticator and the authentication server. Remote Authentication Dial-In User Service (RADIUS) [21] (RFC 2865 – 2869, RFC 3162, RFC 2548) is the most common back-end protocol. IEEE standardized in IEEE 802.11f [7]. It was designed for the enforcement of unique association throughout a ESS (Extended Service Set) and for secure exchange of station's security context between current access point(AP) and new AP during handoff period.

Stemm and Katz [23] defined the idea of vertical handoff. Mishra *et al* [18] analyzed the handoff process at the link layer and found that the factors that influence the handoff latency are probe, authentication, and reassociation delays. They showed that the probe delay is the primary contributor to the overall handoff latency and it is significant enough to affect the quality of service for many applications. Arbaugh *et al* [9] did empirical studies on handoff latencies, supporting our motivation of reducing the re-association delay in a vertical handoff. They also used neighbor graphs and proactive caching algorithms to reduce the reassociation delay.

The IETF Seamoby group [8] defined different protocols for seamless IP-level handoff by reducing network discovery and reconfiguration delays.

Compared with our scheme, they all assumed a *shared AS* between different networks. Bargh *et al* [10] found that IAPP and Seamoby results are not directly applicable for inter-domain seamless mobility and extending them for inter-domain mobility requires enhancements to the security infrastructure.

VII. CONCLUSION

We presented three protocols, SAP v1,v2,v3, in conjunction with the full authentication protocol to deal with vertical handoff across heterogeneous networks integrated in the fashion of loosely-coupled interworking. Our protocols were designed to avoid performing full authentication prior to the handoff. Instead, we used a temporary handoff key (which are K_{STA} in v1,v2, and k_j in v3) to temporarily associate STA with AP while performing the full authentication simultaneously. This facilitates the seamless handoff process because the temporary handoff key bridges the gap caused by full authentication. Simulation results showed that our protocols are efficient and do not have high communication overhead. The security of SAP v1 originated from using different temporary keys for SAP handoff, but this causes more overhead of key management. The efficiency of SAP v2 originated from using a flat key for temporary handoff, but this causes security flaws. SAP v3 was designed by combining v1 and v2 in a randomized fasion, thus providing a complete spectrum of tradeoffs between security and efficiency. We believe that, with the parameter r properly chosen, SAP v3 can provide good security for this temporary handoff process (say 10 seconds), and after that the full authentication results can take over. The randomization was designed in such a way to discourage attackers even for this short period of time. For this we strongly believe that SAP v3 is efficient, secure, and practical in providing fast authentication for vertical handoff.

REFERENCES

- [1] VINT Group, UCB/LBNL/VINT Network Simulator–ns (Version 2). <http://mash.cs.berkeley.edu/ns>.
- [2] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Nov. 1997. *IEEE Standard 802.11*.
- [3] IEEE. Standard for Local and Metropolitan Area Networks – Port-Based Network Access Control. <http://standards.ieee.org/getieee802/download/802.1X-2001.pdf>, 2001. *IEEE Draft P802.1X/D11*.

- [4] IEEE. Draft 4 Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, July 2002. *IEEE Draft 802.11/D4*.
- [5] IEEE. Draft 5 Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, Jan. 2003. *IEEE Draft 802.11/D5*.
- [6] IEEE. Medium Access Control (MAC) Security Enhancements, May 2003. *IEEE Standard 802.11i/D4.0*.
- [7] IEEE. Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, July 2003. *IEEE Standard 802.11f*.
- [8] IETF. Context Transfer, Handoff Candidate Discovery, and Dormant Mode Host Alerting (Seamoby). <http://www.ietf.org/html.charters/OLD/seamoby-charter.html>, June 8 2004.
- [9] W. A. Arbaugh, A. Mishra, M. Shin, N. Petroni, T. C. Clancy, I. Lee, and K. Jang. Using Neighbor Graphs in Support of Fast and Secure WLAN Mobility. <http://www.umiacs.umd.edu/partnerships/ltsdocs/LTS-talk-04-1.pdf>, Feb. 4 2004.
- [10] M. S. Bargh, R. J. Hulsebosch, E. H. Eertink, A. Prasad, H. Wang, and P. Schoo. Fast Authentication Methods for Handovers between IEEE 802.11 Wireless LANs. In *ACM WMASH'04*, pages 50–60, 2004.
- [11] M. Bellare, R. Canetti, and H. Krawczyk. Message Authentication Using Hash Functions: The HMAC Construction. *RSA Lab's CryptoBytes*, 2(1), 1996.
- [12] M. Buddhikot, G. Chandranmenon, S.-J. Han, Y.-W. Lee, S. Miller, and L. Salgarelli. Integration of 802.11 and Third-Generation Wireless Data Networks. In *IEEE INFOCOM'03*, San Francisco, USA, Mar. 30 – Apr. 3 2003.
- [13] J. Edney and W. A. Arbaugh. *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. Addison-Wesley Professional, July 15 2003.
- [14] J. W. Floroiu, R. Ruppelt, D. Sisalem, F. Focus, and J. V. Stephanopoli. Seamless Handover in Terrestrial Radio Access Networks: A Case Study. *IEEE Communication Magazine*, 41(11):110–114, Nov. 2003.
- [15] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. *Journal of the ACM*, 33(4):792–807, 1986.
- [16] H. Krawczyk, M. Bellare, and R. Canetti. Keyed-Hashing for Message Authentication. <http://www.ietf.org/rfc/rfc2104.txt>, Feb. 1997. *IETF RFC 2104*.
- [17] J. McNair, I. F. Akyildiz, and M. D. Bender. An Inter-System Handoff Technique for the IMT-2000 System. In *IEEE INFOCOM'00*, volume 1, pages 208–216, Tel Aviv, Isreal, Mar. 2000.
- [18] A. Mishra, M. Shin, and W. Arbaugh. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. Technical Report UMIACS-TR-2002-75, University of Maryland – College Park, 2002.
- [19] A. Mishra, M. Shin, and W. Arbaugh. Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network. Technical Report UMIACS-TR-2003-46, University of Maryland – College Park, 2003.
- [20] K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Wlianttila, J. P. Makela, R. Pichna, and J. Vallstron. Handoff in Hybrid Mobile Data Networks. *IEEE Personal Communications*, 7(2):34–47, Apr. 2000.
- [21] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote Authentication Dial-In User Service (RADIUS). <http://www.ietf.org/rfc/rfc2865.txt>, 2000. *IETF RFC 2865*.
- [22] W. Stallings. *Cryptography and Network Security: Principles and Practices*. Pearson Education, Inc, third edition, 2003.
- [23] M. Stemm and R. H. Katz. Vertical Handoffs in Wireless Overlay Networks. *ACM MONET*, 3(4):225–350, 1998.
- [24] Q. Zhang, C. Guo, Z. Guo, and W. Zhu. Efficient Mobility Management for Vertical Handoff between WWAN and WLAN. *IEEE Communication Magazine*, 41(11):102–108, Nov. 2003.