Wireless Pers Commun
DOI 10.1007/s11277-007-9271-6

Protected session keys context for distributed session key management

Dan Forsberg

Received: 12 May 2006 / Accepted: 27 January 2007

Abstract Handoffs must be fast for wireless mobile nodes (MN) without sacrificing security between the MN and the wireless access points in the access networks. We describe and analyze our new secure Session Keys Context (SKC) scheme which has all the good features, like mobility and security optimization, of the currently existing key distribution proposals, namely key-request, pre-authentication, and pre-distribution. We analyze these solutions together, and provide some conclusions on possible co-operative scenarios and on which level of the network to implement them. Finally before conclusions we provide some handoff delay simulation results with SKC and key request schemes with corresponding example handoff scenarios with a next generation radio link layer.

Keywords Separate authentication keys · Fast handoffs · Context transfers · Mobility · Pre-authentication · Pre-distribution · Key-request

1 Introduction

Session Key (SK) management for wireless Mobile Nodes (MN) has been an active research topic. IEEE groups like 802.11 Task Group R, 802.21, and 802.16 (WiMAX) are working with issues to improve support for MNs without sacrificing security of their network access sessions. Extensible Authentication Protocol (EAP) working group in the IETF is working with key hierarchies and derivation issues [2]. IETF PANA (Protocol for carrying Authentication for Network Access) and HOKEY (Handover Keying) [10] working groups have been tackling the issue of mobility optimizations [6,7].

One of the adopted requirements in the SK management area is to have cryptographically separate[1] SKs for every Access Point (AP) [9]. There are different proposals to fulfill this requirement, which is important especially in cases, when the encryption of the data packets terminates in the AP, and not deeper in the network. In this paper, we describe our novel Session Keys Context (SKC) scheme, which utilizes a context transfer protocol [14,15,17] between APs. We analyze three existing proposals, namely pre-distribution [3], key-request, and pre-authentication [8], and compare them with our SKC approach.

Our paper is organized as follows. We describe three existing separate SK distribution mechanisms for APs in Sect. 2, and in Sect. 3 we describe our SKC solution. Analysis and comparison between

D. Forsberg (✉)
Nokia Research Center, Helsinki, Finland
e-mail: dan.forsberg@nokia.com

[1] With cryptographically separate keys we mean that the keys do not have a direct key derivation relationship together.

the different alternative solutions is done in Sect. 4 and 5, we provide some results from our simulations with two different handoff scenarios. Finally we conclude our paper.

1.1 Reference Architecture

Our focus is on a reference architecture, where a centralized gateway (GW) and a centralized Key Distributor (KD) are connected to multiple APs (see Fig. 1). The MN connects to the APs via a wireless interface, but also wired interface could be used. The KD takes care of the user authentication and key distribution for the APs and GW routes the MN's data packets. We use this model as our reference architecture, as it is generic enough and reflects the architecture of a local AAA server and an IP packet data GW router.

Our model can be mapped to 3GPP All-IP Network (AIPN) Radio Access Network (RAN) Long Term Evolution (LTE) architecture where the Mobility Management Entity (MME) plays a KD. In WiMAX the Access Service Network (ASN) gateway plays as the gateway router and the Connectivity Service Network (CSN) AAA server as the KD. In a WLAN, the KD could be a local AAA server and the gateway as the normal IP router GW. The mappings can be done differently as well, but the core functionalities of our reference architecture (AAA, routing, and mobility management) must still exist for a mobile access network.

The keys are used in the APs to protect the data packets and signaling between MN and AP. This way the APs can filter packets that are not properly protected and shield the GW from direct attacks with unauthorized packets. Another approach would be to protect the data packets up to the GW, like in a case of an IPSec GW for Virtual Private Network (VPN) connections. However the signaling requires protection between MN and the APs, meaning that key management must be implemented between MN and the APs as well.

The KD has a security association (SA) with all the APs (see SA1, SA2, and SA3, respectively, in Fig. 1). These SAs can be used to integrity protect and encrypt (e.g., *cipher*) signaling packets to preserve confidentiality and integrity of the information between the APs and the KD.

Both the MN and the KD have a common Key Root (KR), formed as a result of an authentication protocol run between the MN and an Authentication Server (AS, see Fig. 1), which contains the long-term credentials for the MNs. Authentication protocol run (for example EAP [1]) is out of the scope of this paper, but we assume that the KR has been established and forms the root of the key hierarchy from the KD downwards to the APs (see Fig. 2).

1.2 Session keys

Figure 2 shows a simplified key hierarchy, where long-term credentials are used to derive a KR as a result of the authentication between the MN and the AS. The KR is used to derive a shared secret SK, between the MN and the currently serving AP. The SK is used to create fresh encryption and integrity protection keys that protect the packets on the wireless link between MN and the respective AP. However, AP may use only integrity protection or encryption or both for different packets depending on the design of the system (for example control and user plane and/or protocol separation).

The SK derivation is a process in which a Key Derivation Function (KDF) is used to create new keys from existing keying material (i.e., KR in this case, but also similarly from SK when creating encryption and integrity protection keys). A KDF is typically based on a one-way hash function. An example of the SK derivation function is given below.

$$\mathrm{SK}_{\mathrm{MN}_X\mathrm{AP}i} = \mathrm{KDF}\{\mathrm{KR} \parallel \mathrm{ID}_{\mathrm{AP}i} \parallel \mathrm{TID}_{\mathrm{MN}_X} \parallel \text{“AP Key”}\}, \quad (1)$$

where i is AP index number, $\mathrm{SK}_{\mathrm{MN}_X\mathrm{AP}i}$ is Authentication Key between APi and MN_X, $\mathrm{KR}_{\mathrm{MN}_X\mathrm{KD}}$ is Key Root between MN_X and KD, $\mathrm{ID}_{\mathrm{AP}i}$ is Public Identity of the APi, $\mathrm{TID}_{\mathrm{MN}_X}$ is Access network specific MN_X identifier, and "AP Key" is a constant string.

Nonces[2] are not used in the SK derivation because the system must be able to derive keys

[2] Random number used once in an authentication protocol in key derivation to prevent replay attacks.

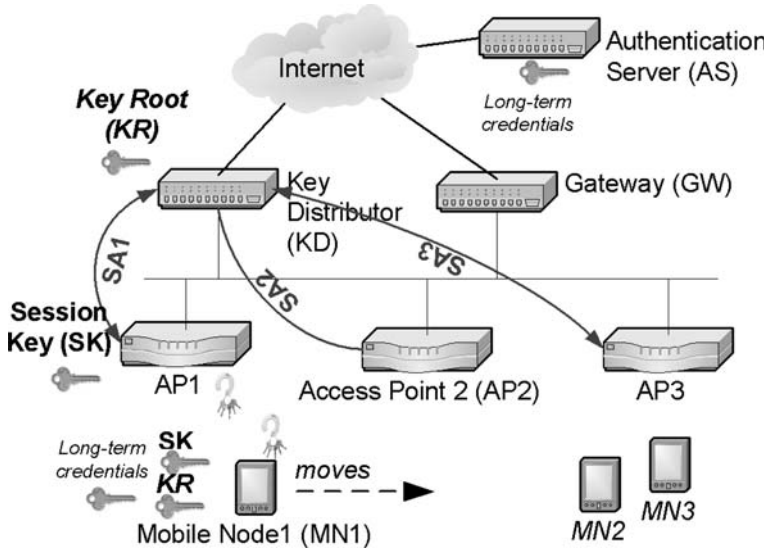


Fig. 1 Reference architecture

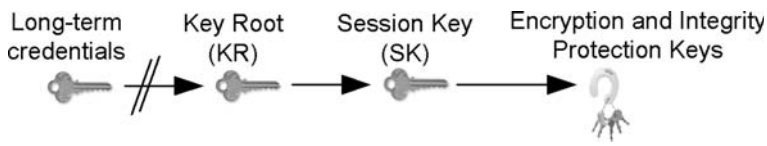


Fig. 2 Simplified key hierarchy

based on existing information. However, encryption and integrity protection key derivation requires them to preserve freshness of the keys (i.e., the same encryption and integrity protection keys are not used twice in the AP). These keys are derived from the SK.

Since MN has the KR it can create $SK_{MNx-APi}$ for each AP*i*, when it knows the AP*i* identity (ID_{APi}). APs themselves do not have the KR, but they need to get the derived key from the KD. SAs between the KD and the APs are used to encrypt the keys for the APs.

1.3 Compromised AP threat

A requirement for the SK management has been created [9] to lower the risk of a compromised-AP (authenticator). The adopted requirement is that the system must have cryptographically separate MN SKs with every AP MN is communicating with. This means that when MN moves from AP_1 to AP_2, it must change its SK. If one AP is compromised the keys for other APs are not compromised and thus the scope of the attack is reduced with cryptographically separate keys.

Based on our reference architecture and the KR between MN and KD we assume that the MN is always able to derive AP specific SKs based on the information from the AP. In high level the KDF is fed with the KR and AP identity information, resulting to a SK bound to the AP's identity. This mechanism to bind the key to its context is called channel binding [2] and is not restricted to only AP identity binding, but can for example include also the used algorithms, multiple identities, etc.

The MN gets the AP identity for example during the handoff (HO), i.e., when it knows which AP to connect to. During the HO the MN may also need to send its identity to the AP, so that AP is able to find the right SK for the MN.

The schemes where the same SKs (common keys approach) are used in multiple APs are not considered in this paper, because they do not fulfill the adopted security requirement of separated SKs. However, the threat of a compromised AP becomes less interesting if the data traffic encryption is terminated deeper in the network than in the AP. Separate keys can be used to improve the security of the HO signaling and binding key derivation to the AP identity prevents for example false AP attacks as the APs are authenticated based on the

SKs. Because of cryptographically separate keys the MN can send a signed messages to the target AP via the source AP, which the target AP can verify. This means that APs cannot execute HOs without MN's involvement and thus launch DoS attacks against each other based on HO signaling.

Key derivation mechanisms between APs and the MN that use public (asymmetric) key cryptography, like AP certificates, fulfill the requirement of cryptographical SK independence, but typically require heavier computation than symmetric key cryptography (shared secrets) and are thus not used for time critical HO processing.

2 Existing authentication key management mechanisms

In this chapter, we describe three existing proposals for mobile networks that provide separate MN SKs for APs, namely pre-distribution, key-request, and pre-authentication.[3]

In the pre-distribution (or pre-emptive keying) [3,11,18,19] scenario (Fig. 3a) the KD derives AP specific SKs and distributes them to a number of APs when MN has successfully attached to the access network. Channel binding mechanism is used in AP specific key derivation. The number of APs included in the pre-distribution scheme can vary (i.e., certain group of APs).

In a key-request scenario (Fig. 3b) the KD delivers a new key to the AP during each HO. When MN moves from AP_1 to AP_2, it authenticates through AP_2 to the KD. KD derives a MN specific SK and delivers it to the AP based on the request coming from the AP or MN. This scenario is simple and efficient in cases where the HO signaling goes through a centralized element also providing the KD functionality (for example a WLAN switch).

In a pre-authentication [8] scenario (Fig. 4a) the MN authenticates to multiple APs through a single AP [4,12,20,21]. This way the MN can pre-establish SKs with multiple neighboring APs.

[3] We have omitted Kerberos [22], because it requires the MN to carry the keys for the servers (i.e., APs in this case).

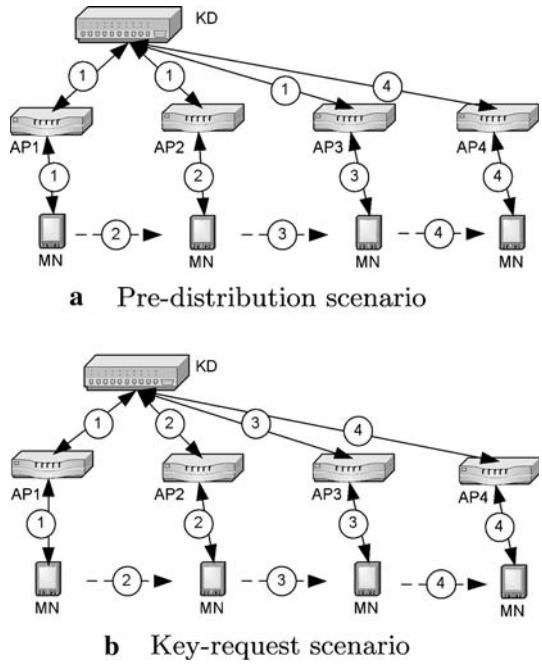


Fig. 3 Pre-distribution and key-request scenarios

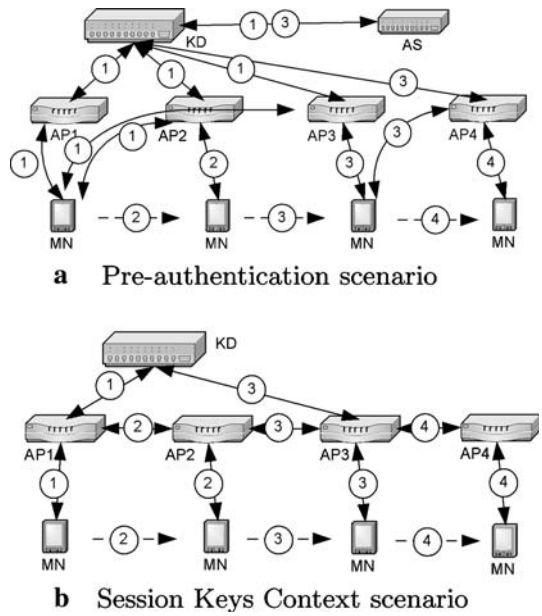


Fig. 4 Pre-authentication and SKC scenarios

3 Our new session keys context concept

We have created a new way to distribute keys to the APs in a SKC. SKC contains multiple SKs separately encrypted for different APs. The SKC is

Table 1 An example SKC row for two APs

AP Id	Encrypted SK	MAC
ID_{AP_1}	$E_{SA_{AP_1}}\{SK_{MN_X AP_1}\}$	$MAC_{SA_{AP_1}}\{ID_{AP_1} \parallel E_{SA_{AP_1}}\{SK_{MN_X AP_1}\}\}$
ID_{AP_2}	$E_{SA_{AP_2}}\{SK_{MN_X AP_2}\}$	$MAC_{SA_{AP_2}}\{ID_{AP_2} \parallel E_{SA_{AP_2}}\{SK_{MN_X AP_2}\}\}$

created in the KD for a number of APs and sent to the AP that the MN is currently attached to. When MN moves, the SKC is transferred between APs. Transferring the SKC between APs may utilize for example the context transfer protocol [17], or inter access point protocol [13]. Each AP gets the SK from the SKC and creates encryption and integrity protection keys from it, similarly to the other approaches. Figure 4b shows the SKC scenario. The motivation to use context transfer between APs is to reduce the signaling load with the KD and to make KD independent of the time critical mobility signaling, which allows more relaxed KD placement in the network.

The SKs are encrypted and accompanied with AP identity information. The encrypted SK and AP identity information are signed using a SA between the KD and the AP. Each AP that receives this context finds its own encrypted SK based on its identity. Below (see Table 2) is an example context row (SKC). The row is protected with Message Authentication Code (MAC) [16].

In case the SKC does not contain a SK for an AP, the KD must update the SKC. This is a similar situation to the pre-distribution case, where the MN crosses the pre-distribution area border. However, the AP may also contain information about its neighboring APs and request an update from the KD before the MN actually moves to an AP, that is, not included in the SKC. Step 3 in Fig. 4b highlights this scenario, where AP_3 notices that the SKC does not contain a row for AP_4 and requests a context update from the KD. When MN moves from AP_3 to AP_4, both the original context rows and the updated context row are transferred from AP_3 to AP_4 in one SKC (step 4). Alternatively, the target AP_4 can request new SKC from the KD.

3.1 Extending SKC for AP to AP signaling security

The SKC row could also contain keying material used to create signaling and context transfer protection keys between the APs. In this case the KD creates a Signaling Protection Key (SPK) and inserts this key into all the SKC rows together with the AP specific SK making it available for each of the APs separately.

When the APs want to communicate with each other securely they can use the SKC rows to get the SPK and further derive encryption and integrity protection keys for the AP–AP communication or just use the SPK. Depending on how this happens, the same keys could be used between multiple APs listed in the SKC. The SKC concept can be used to provide fresh enough keying material for the AP–AP HO related communications, since with the HO signaling there is also SKCs involved.

Even if the SPK is same for all APs in the SKC, it is not the same between different MN's SKCs. This makes it possible for the APs to choose between different SKCs, meaning that the key used to derive encryption and integrity protection keys for AP–AP communication is not necessarily the same between other APs.

4 Analysis of the SKC concept

The APs only need to keep SKC objects for the MN's they are serving currently. Unused objects can be removed. Thus, AP resources are dependent only on number of concurrent MNs an AP is serving at any moment. If the SKC grows too big for a single AP to handle it can cut rows from the SKC structure (for example keys that are not for its neighbor APs).

The MN does not have to send the SKs to the APs or know the APs included in the SKC. Space requirement in the MN is not bound to the number of APs because the SKs (SK_{APi}) are derived from the Root Key (RK).

Implicit AP authentication (via KD) without the need to have AP specific certificates. If the AP is able to decrypt messages with the proper SK, then the AP is validated (proof of key possession).

The SKC construction allows the context to be sent in multiple separate packets. The old AP knows (ID_{APi}) the entry in the SKC structure that contains the SK for the target AP. Old AP can send this entry first to the target AP together with other high-priority context data. This makes the signaling delay less dependent on the SKC size itself if needed.

The SKC scenario can efficiently use the available memory in the APs to reduce signaling load in the KD. The more memory used, the less signaling with the KD. For the encryption of the keys a common initialization vector can be used for all rows, even different keys are used (must make sure that the same IV is not used twice).

The AP can request updates from the KD, meaning that the KD does not have to push new rows for the SKC. This is a benefit if the server (KD) initiated signaling is seen problematic. For example if the KD does not know in which AP the MN currently is.

The AP knows the scope of the SKC, since it can look which APs are included in it. But the AP may not know the topology of the network and/or its neighboring APs. However, the SKC could also contain coded embedded network topology information, which could provide the needed information for the APs.

The SKC concept is flexible in terms of what information is put into the SKC. The AP could even add information to its row in the SKC itself, encrypting parts of the row and then integrity protecting the whole row again. An AP is only able to do this for its own row, not for the other APs, because the SAs are AP specific. AP could for example add accounting information related to the MN. This way the APs could aggregate information into the SKCs and send this information back to the KD for further processing if needed. Instead of updating the session information like location and accounting records periodically to the KD, the information can be securely stored into the SKC structure, regardless of the mobility frequency. This provides the same benefit as the SKC was designed for, i.e., reducing for example accounting related signaling load by using more memory. This model of accounting could be feasible for example for a capped flat rate model. The SKC becomes a structure of the MNs context, containing session specific information.

The benefit of the SKC is also that the SKs can be stored into the APs memory in encrypted form together with the MACs that provide integrity protection and thus also authentication of the SKC row. If the SKC row decryption, verification, and encryption and integrity protection key derivation processes are implemented in a secure hardware chip (e.g., Trusted Platform Module) it is more difficult for the attacker to get the SKs by just examining the SKC in the AP's memory or on the wire between APs.

The KD needs to know for which APs it will create the SKs. However, the KD may also optimize the contents of the SKC to include APs that are more commonly used (based on mobility pattern analysis), or creating SKCs for certain groups of APs or simply for all APs it controls. KD could also let the AP request the new keys for other APs if the KD does not know the topology of the APs. Selecting the scope of the SKC can be a configuration parameter reflecting physical and geographical location of the APs and subscribers. Creating user subscription specific SKC is possible, which allows subscription specific network coverage, for example "hot-spots area," "city wide access," or optionally also based on subscriber's mobility pattern like "John's home and work route." Further details are out of scope of this paper.

Integrity protected or encrypted packets in the MN's buffers need to be discarded when AP changes (new keys). However, this is valid for all approaches, where different keys are used for APs, and thus is not only a SKC disadvantage.

A logical interface between APs is needed for the context transfers. This is a general requirement for all systems that have AP–AP communication. However, the benefit the SKC concept is that in the scope of the SKC, regardless of how frequently the UE changes the AP, there is no need to signal with the KD. This provides better scalability for the KD.

4.1 SKC Memory requirements

The total amount of memory required in an AP with the SKC concept depends on (1) the number

of concurrent and active MNs attached to it (number of SKCs), (2) the size of the SKC structure, and (3) the number of APs (rows) included in one SKC.

The structure size in the SKC depends on the size of the MAC, SK, and the AP ID.

We have assumed that the encryption does not add to the size of the SKC row as the structure can use a common initialization vector (IV) if needed. Thus, the amount of memory required in one AP can be calculated roughly with the formula below (2), where s is a function returning the size in bytes of the structure provided as an argument (e.g., *sizeof()*). $\mathrm{nSK_{max}}$ is the max number of rows in the SKC and $\mathrm{nMN_{max}}$ the maximum number of concurrent and active MNs in the AP.

$$M = s(\mathrm{IV}) + n\mathrm{SK_{max}} * n\mathrm{MN_{max}} * (s(\mathrm{MAC}) + s(\mathrm{SK}) + s(\mathrm{ID_{AP}}i)). \quad (2)$$

Additional protection is needed when the context is transferred from the KD to the AP. Also the MN's identity is needed if the SKC can not be identified otherwise.

If we specify that the MAC is 160 bits long (SHA-1 for example), key size 128 bits (considered to be secure today) and AP ID 32 bits long (length of an IPv4 address), then one row is 40 bytes (320 bits). SKC that contains keys for 30 APs is then 1,200 bytes and still a way below the usual size of an MTU in Ethernet. Further more, an AP containing SKCs with 30 keys each and for 20 MNs would consume about 24 k bytes for the SKCs only.

In city wide WLAN AP networks, where the number of APs may be even hundreds or thousands, the SKC concept could be used to provide keys for all the APs at the same time, making the signaling between the KD and APs unnecessary during the SK lifetime. In this case the APs do not have to care about the neighboring APs and check if the SKC contains keys for them. If the SKC would contain keys for 300 APs, the memory requirement in an AP would be about 12 kbytes per MN.

4.2 SKC Concept comparison with pre-distribution

In the SKC concept the keys are bundled into a data structure (SKC) and sent to the MN's current AP. However, the MN may not visit all the APs provided in the SKC. The pre-distribution scheme is similar in this sense that the AP may not use the keys it has received from the KD. Actually, the SKC concept could be seen as a signaling optimization of the pre-distribution scheme. Thus, SKC scheme scales the better the more APs are involved when comparing it to the pre-distribution scheme. In the pre-distribution scheme the KD needs to send the key separately to all the APs and also receive an acknowledgment from them, involving all the APs. In the SKC concept this signaling is not necessary.

In the pre-distribution scheme the APs are involved in signaling with KD every time a new MN attaches to the area of the APs. This means that the APs need more control signaling capacity and state information than in the SKC scenario. More signaling capacity is needed from the KD as well.

In our SKC approach the required memory is per AP not for all APs simultaneously, but the overall memory consumption in the system may be slightly bigger with the SKC, since the AP needs to store the MACs and AP identities together with the SKs in the SKC.

Pre-distribution scheme is fast from the HO point of view, since the key is already located in the target AP. However, the AP needs to find the correct key and needs an identifier from the MN before this can happen. AP-to-AP level signaling could be used to inform the target AP about the MN also in the pre-distribution case. However, in our paper we assume that the pre-distribution scenario does not require AP–AP interface. In our approach, if the context transfer can happen proactively the target AP can prepare itself before the MN really attaches to it. However, with the SKC scheme the APs could store the received SKCs into a cache memory and check if they already have an SKC for the MN that attaches to the AP. The benefit of this mechanism depends on the handoff signaling design. Pre-distribution of the SKC entries from serving AP to neighboring APs is also possible as described above.

In pre-distribution scheme all the APs included in the pre-distribution area know at least the number of MNs attached to the area. If attacker has gained access to the AP, he may know how many MNs are active in the pre-distribution area or even

identify the MNs, depeding on how the keys are identified and named. However, the pre-distribution area border may not be visible in the AP itself as it can be controlled by the KD. Advantage over pre-distribution scheme with the SKC is that a possible rogue-AP does not know which MNs are attached to the pre-distribution area.

Also, in pre-distribution scheme the KD must know for which APs to send the keys and when to send more keys if the KD did not send the keys to all the existing APs under its control. This applies to the SKC as well in the form of selecting APs into the SKC and updating the SKC when necessary. This responsibility can, however, be delegated to the APs with SKC scenario, so that the APs are aware of the network topology around them and request new keys from KD.

4.3 SKC Concept comparison with key-request

Key-request method requires HO synchronized signaling with the KD during the HOs, which makes the scheme potentially slower than for example the proactive SKC transfer if the KD is far from the APs. On the other hand no inter-AP interface is needed with the key-request scheme. If the inter-AP interface is not available due to physical transport topology or protocol reasons, key-request mechanism could be considered roughly as fast as context transfer through the KD. In this case however the key-request requires key derivation in the KD, while the SKC transfer does not need that but requires a bigger packet (SKC versus one SK).

Key-request method does not require as much memory from the APs as the SKC method, and also the keys are newer and created only when actually needed. Thus, we can say that the key-request method is AP memory optimized but with the cost of signaling load between APs and KD, whilst the SKC concept is signaling load optimized with the cost of memory and packet size. The SKC also makes the HO procedure independent of the delay between the APs and the KD. This improves the overall scalability of the network.

With SKC the load on the KD is independent of the AP–AP handover frequency and also partly independent of the number of APs in the network. The same is not true with the key-request scheme, because it is involved with every HO regardless if the MN switches between the same APs all the time. This is a considerable advantage for the SKC scheme, but applies also for the pre-distribution and partially also pre-authentication schemes.

Key request could also be seen as a special case of the SKC scenario, where the SKC includes only one key at a time. This would mean that either the source or target AP requests SKC update from the KD during a HO, similarly to the key request scenario.

4.4 SKC Concept comparison with pre-authentication

In pre-authentication the KD and AS are involved multiple times because the MN authenticates to multiple APs through one AP. This is needed, because the pre-authentication signaling is considered to take much more time than is generally applicable during the HO process. Also, if handover decisions happen very fast it may be possible that the MN has not pre-authenticated to the target AP yet. To partly fix this issue the MN should pre-authenticate with all the neighboring APs. However, the MN may not have time to do this, when a new handoff must occur. This scheme is not suitable for AP–AP level handoffs for fast moving MNs if the pre-authentication signaling must traverse a long path to the authentication server (delay and authentication server scalability). The neighboring APs also need to reserve memory for the keys, even if the MN never attaches to all of the APs.

This scheme may become attractive for higher level HO that happen for example between the KDs higher up in the network topology. If used this way, this scheme could be combined with the key pre-distribution, key-request, and SKC schemes.

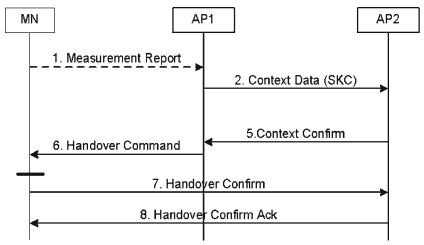
The comparisons have been summarized in Table 2.

5 Total handoff delay simulations with SKC and key-request scenarios

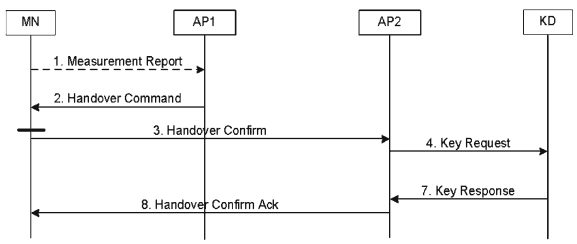
We ran a simple NS-2 (with Mobiwan extension for topology generation) simulation of two differ-

Table 2 SKC comparison with pre-distribution, key-request, and pre-authentication schemes

	SKC	Pre-distribution	Key-request	Pre-authentication
APs involved per MN	1	n	1	m
AP memory per MN	k	1	1	m
AP coverage	k	n	1	m
Control point	AP, KD, both	KD	AP, KD	MN
AP–AP interface	Yes	No	No	No
Key "hit-rate"	−−	− − −	100%	−
HO freq. relation	+++	++	−−	−
HO synch. with KD	+++	++	− − −	+
KD signaling load	+++	++	−−	−−
Complexity	− − −	−−	+++	−
Flexibility	+++	++	−	−



a SKC with prepared handoff



b Key-request with non-prepared handoff

Fig. 5 Handoff scenarios

ent HO scenarios, where the scenario 1 (Fig. 5a) illustrates a prepared HO between APs without KD involvment and scenario 2 (Fig. 5b) illustrates a non-prepared HO with KD involvement. For scenario 1 the SKC scheme was used and the size of the SKC size varies in the simulation (Fig. 7a). In scenario 2 there is no context transfer between APs, but instead a key-request message from target AP to KD and a reply. In both scenarios the number of signaling messages is the same. In scenario 1 only the SKC is transferred between the APs and in scenario 2 the SK is fetched from the KD. AP-KD delay varies in the scenario 2 simulation (Fig. 7b).

For the link layer between MN and APs we used EURANE [5]. EURANE does not support radio

Table 3 Simulation settings

Radio resynch delay	6 ms
MN processing delay	6 ms
AP processing delay	2 ms
KD processing delay	1 ms
AP-AR link bandwidth, prop. delay	100 Mbps or 1 Gbps, 0.125 ms 1 Gbps, variying propagation delay
AP-AR link bandwidth, prop. delay	100 Mbps or 1 Gbps, 0.125 ms 1 Gbps, variying propagation delay

link level handovers, which makes the simulated results more simplistic. Synchronization with the target AP is assumed to take 6 ms (radio break delay). Table 3 shows the configuration settings for the "UMTS" link layer for NS-2. The values are estimates for a next generation high-bandwidth radio. Table 4 shows network node processing delay estimates, link delays and bandwidths for backhaul links. The simulation time for each measurement in NS-2 was 200 s. The results did not noticeably change with longer time periods. Also, the radio link needs to be established before the MN moves to the target AP. This is easy within scenario 1, where the SKC is pushed to the target AP. In scenario 2 preparing the target AP is not possible, without an explicit signal from source AP to the target AP (not shown on the Fig. 5b).

Network topology for the simulation is shown in Fig. 6. When comparing this to the reference architecture (Fig. 1), we have added routers into the topology. The MNs move arbitrarily and HOs are initiated based on MN's signal strength measurements. The context is transferred from AP to

Table 4 NS-2 configuration settings for a next generation radio link in the simulation

Link layer type	"UMTS/RLC/AM"
MAC type	"Mac/UMTS"
Downlink bandwidth	100 Mbs
Uplink bandwidth	50 Mbs
Downlink TTI	0.5 ms
Uplink TTI	0.5 ms

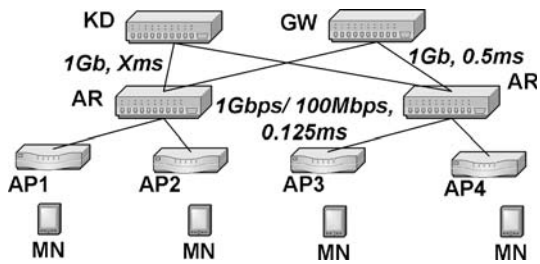


Fig. 6 Simulation network topology

AP via the paths in the figure. For example in a HO from AP_1 to AP_2, the context is sent through an Access Router (AR), and in case of AP_2 to AP_3 the context goes through an AR and a GW. This affects the total HO time in case of scenario 1, where signaling goes between APs and context size varies. Figure 7a shows the total HO delays for the scneario 1. We ran the two simulations with 1 Gbps and 100 Mbps links, respectively, from APs to the ARs for the SCK scenario. The links were specified as duplex-links and fragmentation was not implemented. Size for the signaling messages on the backhaul links was around 100 bytes for all messages except for the context transfer messages (e.g., varying SKC size).

In scenario 2 the link between the ARs and the KD varied from 0.1 to 2.3 ms. In our simulations we did not use any background traffic.

5.1 Simulation results

The SKC transmission delay increases the total HO time, but the delay increase is stepwise (each step roughly 0.5 ms), because of the 0.5 ms Transmission Time Interval (TTI) value on the radio link. With high-bandwidth links (like 1 Gbps) behind the AP, the size of the SKC does not add considerably to the total HO delay. With a 100 Mbps duplex link the size of the context affects more than with 1 Gbps, which is natural as higher bandwidth

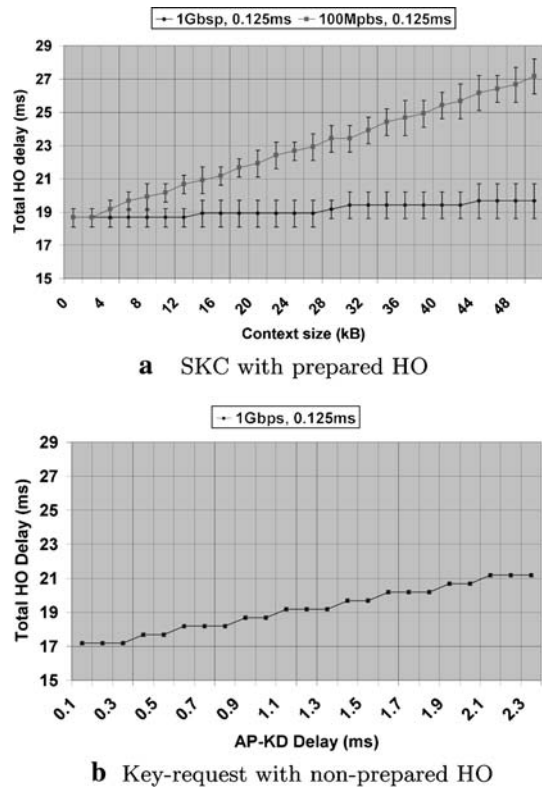


Fig. 7 Total handoff delays

links carry bigger packets faster than lower bandwidth links. With small SKC sizes (below 2 kb), the increase into the total HO delay is not in average meaningful. The steps in the curves depend on when the next time slot for the radio transmission happens to be. For a 100 Mbps link a 12 kb context size (separate keys for 300 APs) adds about 2ms more delay to the total HO delay, whilst on a 1Gbps link 50 kb increase in the context size adds 1ms to the total HO delay. With the possibility to transfer the target AP specific SKC row first during the HO, the size of the SKC does not pose any restrictions or remarkable delays to the HO procedure. However, the bigger the SKC, the more data needs to be transferred between the APs.

When comparing the total HO delays together with scenarios 1 and 2, we can conlude that scenario 2 is faster than scenario 1 if the delay between AR and KD is below 1 ms, regardless of how big or small the SKC is in scenario 1. The difference between the scenarios depends on the network topology (hops between the APs versus Hops between

APs and KD), transmission and propagation delays, and size of the transferred data (SKC transfer versus SK transfer; signaling between APs vrs between AP and KD). In the total HO delay of 20 ms an additional 1 or 2 ms delay (5–10%) is not noticeable, especially if MN's user data packets are not lost during the HO. As an example, forwarding packets from the old AP to the new AP when MN has left the old AP and the old AP still gets packets for the MN is one way to provide lossless packet delivery for MNs.

6 Conclusion

We introduced a novel SKC concept that provides cryptographically separate AP SK for the MN without involving the KD during the HOs, and not involving APs that do not serve the MN. We analyzed and compared the SKC concept with three other existing key distribution schemes, namely key-request, key pre-distribution, and pre-authentication.

The SKC concept optimizes the signaling load in the KD and lessens the real-time requirements for it, thus making it more scalable for serving more APs in the network and making the transmission delay between the APs and the KD not a factor during HO. The drawback with this concept is that there needs to be a logical interface between the APs. However, this is not a problem from deployment perspective in IP networks, but may bring additional burden for interoperability between different vendors if not standardized.

We conclude that the key-request scheme applies well for cases when the KD resides close to the APs and the implementation is based on hardware (simpler mechanism to implement). This is the case for example with WLAN switches. However, in a city wide AP network, there needs to be quite many switches, which then may need physical security as well because they possess the root session keys (KR). SKC concept suits better for cases, where the KD is higher up in the network serving more APs. This way it is also easier to protect KD physically (location). For example an AAA proxy or server could implement the SKC functionality. However, for the SKC update procedures and error cases the key-request scheme may have to be used between the APs and the KD. SKC context size does not affect much the total HO delay as it is not transferred over the radio link between MN and APs, but over high-speed wired links between APs. For lower bandwidth links between APs, the APs can send only part of the SKC to make the HO delays in acceptable boundaries if needed. Thus, we think that SKC as a concept is very scalable and suits nicely for flat architectures, where HO signaling happens between APs and key management happens deeper in the core network.

The pre-authentication scheme does not suit for AP–AP HOs, but can be used for HOs that happen in the higher layer, for example between KDs making it a natural companion with the SKC concept with multiple KDs.

Acknowledgments We would like to thank Lauri Tarkkala for taking part designing the SKC scheme and providing valuable feedback. We thank Guo Wei, Hui Huang, and Dongmei Zhanhong for providing the basic NS-2 framework for the simulations.

References

1. Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., & Levkowetz, H. (2004). Extensible authentication protocol (EAP). RFC 3748 (Proposed Standard). [Online]. Available: http://www.ietf.org/rfc/rfc3748.txt.
2. Aboba, B., Simon, D., Arkko, J., Eronen, P., & Levkowetz, H. (2006). Extensible authentication protocol (EAP) key management framework. Internet-Draft (work in progress), draft-ietf-eap-keying-15.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-eap-keying-15.txt.
3. Arbaugh, W., & Aboba, B. (2003). Handoff extension to RADIUS. Internet-Draft (work in progress, expired), draft-irtf-aaaarch-handoff-04. [Online]. Available: http://tools.ietf.org/html/draft-irtf-aaaarch-handoff-04.
4. Balfanz, D., Smetters, D. K., Stewart, P., & Chi Wang, H. (2002). Talking to strangers: Authentication in ad-hoc wireless networks. *In symposium on network and distributed systems security* (NDSS '02). San Diego, CA. http://citeseer.ist.psu.edu/balfanz02talking.html.
5. European Commission 5th framework project SEACORN (http://seacorn.ptinovacao.pt/), "EURANE - Enhanced UMTS Radio Access Network Extensions for NS-2," referenced 2006-11-30. [Online]. Available: http://www.ti-wmc.nl/eurane/.
6. Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., & Yegin, A. (2005). PANA mobility optimizations. Internet-Draft (work in progress), draft-ietf-pana-mobopts-01.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-pana-mobopts-01.txt.

7. Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., & Yegin, A. (2006). Protocol for carrying authentication for network access (PANA), Internet-Draft (work in progress), draft-ietf-pana-pana-12.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-pana-pana-12.txt.
8. Hoeper, K., & Gong, G. (2003). Models of authentication in ad hoc networks and their related network properties. Department of Electrical and Computer Engineering University of Waterloo, CACR, Technical Report.
9. Housley, R., & Aboba, B., AAA key management. Internet-Draft (work in progress), draft-housley-aaa-key-mgmt-04.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-housley-aaa-key-mgmt-04.txt.
10. IETF Working Group, "Handoff Keying (hokey)," (2006). [Online]. Available: http://www.ietf.org/html.charters/hokey-charter.html.
11. Institute of Electrical and Electronics Engineers, "802.11r: Transition Acceleration Protocol (TAP)," IEEE, proposal 802.21-04/xxxr1, 2004.
12. Institute of Electrical and Electronics Engineers, "802.11r: Justintime reassociation (jit)," IEEE, proposal 802.21-04/xxxr1, 2004.
13. Institute of Electrical and Electronics Engineers, "Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation," IEEE, Tech. Rep. IEEE 802.11F, 2003.
14. Kempf, J. (2002). Problem description: Reasons for performing context transfers between nodes in an IP access network. RFC 3374 (Informational). [Online]. Available: http://www.ietf.org/rfc/rfc3374.txt.
15. Koodli, R., & Perkins, C. (2001). Fast handovers and context transfers in mobile networks. *ACM SIGCOMM Computer Communication Review, 31*(5), 37–47. ISSN: 0146–4833.
16. Krawczyk, H., Bellare, M., & Canetti, R. (1997). HMAC: Keyed-hashing for message authentication. RFC 2104 (Informational). [Online]. Available: http://www.ietf.org/rfc/rfc2104.txt.
17. Loughney, J., Nakhjiri, M., Perkins, C., & Koodli, R. (2005). Context transfer protocol (CXTP). RFC 4067 (Experimental). [Online]. Available: http://www.ietf.org/rfc/rfc4067.txt.
18. Mishra, A., Shin, M., & Arbaugh, W. (2004). Pro-active key distribution using neighbor graphs. *IEEE Wireless Communications Magazine*, *11*(1), 26–36.
19. Mishra, A., Shin, M., Arbaugh, W., Lee, I., & Jang, K. (2003). Proactive key distribution to support fast and secure roaming. IEEE 802.11 Working Group, Tech. Rep. IEEE-03-084r1-I. [Online]. Available: http://www.ieee802.org/11/Documents/DocumentHolder/3-084.zip.
20. Pack, S., & Choi, Y. (2002). Pre-authenticated fast handoff in a public wireless LAN based on IEEE 802.1x Model. In *Proceedings of the IFIP TC6 personal wireless communications 2002*. In proceedings of the *IFIP* TC6/WG6.8 Working Conference on personal wireless communications, *IFIP* Conference Proceedings, vol. 234, 175–182.
21. Pack, S., & Choi, Y. (2002). Fast inter-AP handoff using predictive-authentication scheme in a public wireless LAN. *Networks* 2002 (Joint ICN 2002 and ICWLHN 2002). Atlanda, USA, August 2002, pp. 15–26
22. Steiner, J. G., Neuman, C., & Schiller, J. I. (1988). Keyberos: An authentication service for open network systems. In *USENIX conference proceedings* Winter 1988, pp. 191–200, http://citeseer.ist.psu.edu/steiner88kerberos.html.

Dan Forsberg (dan.forsberg@nokia.com) received a M.Sc. in Computer Science (Software Engineering) in 2000 from Helsinki University of Technology (HUT), wher he was also a member of the HUT - Dynamics MIPv4 research team. He has background in Unix/Linux system and network administration as well as service provisioning for several years. Currently Dan is working in the Nokia Research Center (2000-) as a Senior Research Engineer in Helsinki. His areas of interests include multi-access mobility, networking protocols, network access security, applied security, security engineering, AAA frameworks, network architectures etc. Mr. Forsberg has been an active member of IETF and 3GPP.