SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks. 2010; **3**:4–15
Published online 8 September 2009 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.135

On predictive routing of security contexts in an all-IP network‡

Hahnsang Kim*,† and Kang G. Shin

Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109-2121, U.S.A.

Summary

While mobile nodes (MNs) undergo handovers across inter-wireless access networks, their security contexts must be propagated for secure re-establishment of on-going application sessions, such as those in secure mobile internet protocol (IP), authentication, authorization, and accounting (AAA) services. Routing security contexts *via* an IP network either on-demand or based on MNs' mobility prediction, imposes new challenging requirements of secure *cross-handover* services and security context management. In this paper, we present a *context router* (CXR) that manages security contexts in an all-IP network, providing seamless and secure handover services for the mobile users that carry multimedia-access devices. A CXR is responsible for (1) monitoring of MNs' cross-handover, (2) analysis of MNs' movement patterns, and (3) routing of security contexts ahead of MNs' arrival at relevant access points. The predictive routing reduces the delay in the underlying security association that would otherwise fetch an involved security context from a remote server. The predictive routing of security contexts is performed based on statistical learning of MNs' movement pattern, gauging (dis)similarities between the patterns obtained *via* distance measurements. The CXR has been evaluated with a prototypical implementation based on an MN mobility model on a grid. Our evaluation results support the predictive routing mechanism's improvement in seamless and secure cross-handover services by a factor of 2.5. Also, the prediction mechanism is shown to outperform the Kalman filter-based method [13] as a Kalman Fiter-based mechanism up to 1.5 and 3.6 times regarding prediction accuracy and computation performance, respectively. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: secure seamless handovers; selective predictive routing; edit distance; χ^2-distance

1. Introduction

Inter-wireless technologies, ranging from IEEE 802 networks such as Wi-Fi, WiMax, and personal area networks, to non-802 networks such as cellular networks, are rapidly converging. This trend has led mobile users to carry multimedia-access devices that operate across heterogeneous networks, without disrupting on-going sessions. Meanwhile, the IEEE 802.21 Standard [2] puts into practice the ability of *vertical* and *horizontal* handovers between domains with different management policies, referred to as *cross-handovers*.

*Correspondence to: Hahnsang Kim, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109-2121, U.S.A.
†E-mail: hahnsang@eecs.umich.edu
‡A subset of this paper was presented at IFIP/IEEE International Symposium on Integrated Network Management 2009 [1].

Furthermore, mobile applications impose new challenging requirements of reducing additional delays [3], e.g., fast key establishment between the involved entities for secure network accesses. A 'security context' that contains such information is essential to the fast and secure re-establishment of the involved security protocol flows as mobile nodes (MNs) cross domain boundaries. In particular, an MN's security context is routed to a target point of attachment (e.g., an access point (AP)) ahead of its arrival, thereby avoiding the disruption of on-going sessions. Security contexts of this kind vary with the corresponding applications, ultimately requiring an effective, integrated, and scalable way of managing security contexts.

1.1. Related Work

A network-layer-based protocol, called context transfer protocol (CXTP) [4], is specified for the purpose of routing (security) contexts. CXTP provides an option for coping with *seamless* handovers of MNs that are equipped with inter-wireless technologies, in conjunction with the IEEE 802.21 Standard. At the same time, maintaining on-going application sessions without disruption requires a 'map' *via* which to make the corresponding security contexts available to a target AP before the MN's arrival at that AP. The map can be represented by a neighbor graph [5] that exhibits the APs' logical connectivity, based on MNs' paths. Also, Chou and Shin [6] proposed a packet buffering-and-forwarding mechanism for smooth handovers. Additionally, Song *et al.* [7] presented a case study that quantifies the possible gains with mobility prediction in a voice over internet protocol (VoIP) application. Similarly, various techniques for the accurate prediction of the MN's future location have been studied. For instance, user mobility profile [8,9], road topology knowledge [10], and MNs' positioning information [11] have been instrumental in the mobility prediction for service provisioning and bandwidth reservation in cellular networks. Despite this extensive research, there is still a gap to fill between the prediction and the efficient routing mechanisms, especially for managing security contexts independently of inter-wireless access networks to enable seamless secure, handover services.

1.2. Challenges

There are two main challenges in developing an integrated framework that enables real-time applications to run without any disruption even when the users undergo handovers. First, such a framework should be able to route security contexts to the corresponding 'receivers', e.g., routers, ahead of the MN's arrival, while keeping the overhead of the routing to a minimum. An effective prediction mechanism can track an MN and then estimate its future direction. To this end, the correct extraction of features from the MN's movements is of utmost importance for prediction. Second, a framework should be easy to deploy. In other words, its deployment should not require any new infrastructure. One way to do so is to build the framework upon the standard technologies.

1.3. Contributions

The contributions of this paper are 4-fold as follows:

- We present a comprehensive framework, the context router (CXR) that monitors, detects, and analyzes the MN's movement and cross-handover. A CXR consists of monitoring, analysis, and routing components. The monitoring component observes and detects the likelihood of the MN's cross-handover, and then tracks directional changes in the MN's movement, yieding the pattern of the association with APs. The analysis component gauges the (dis)similarity between an observed pattern and *a priori* established patterns bound to a distribution of CXR candidates. The analysis ultimately narrows the candidates down to a few. The MN's security context is routed to the selected candidates.
- The abstraction of the MN's movement at the AP level provides a tradeoff between the accuracy in prediction and tracking and the computation overhead. The CXR logs directional changes in association with APs during the MN's movement, generating a *time-history vector of angles* (TVAs). The TVA is an abstraction of the MN's movement pattern where we focus on the MN's association with APs rather than its location.
- An overlay formed by multiple CXRs is independent of the underlying access networks. The overlay gradually learns its topology and, hence, facilitates the deployment of CXRs. The deployment of even only a few CXRs demonstrates the efficacy of predictive routing, achieving scalability and manageability.
- The prototype implementation results in the execution of mobility monitoring, analysis, and filtering. We simulate the MN's movement on a grid, according to a specified mobility model, and then gauge (dis)similarity of the MNs' movement patterns against a database of movement patterns, using the

DOI: 10.1002/sec

χ^2-distance or edit distance [12]. The accuracy of the CXR's prediction mechanism is evaluated, compared with a Kalman filter-based estimation [13].

1.4. Organization

The rest of this paper is organized as follows. Section 2 provides the background of a context-transfer protocol, the concept of security contexts and applications using them, along with motivations. Section 3 describes a mobility model describing the MN's movement. Section 4 describes the design of the CXR that consists of the monitoring of the MNs' movement patterns and cross-handovers, the analysis of movement patterns, and two routing methods, i.e., predictive and reactive routings. Also, implementation issues are discussed. Section 5 evaluates the accuracy of our prediction mechanisms. We conclude the paper in Section 6.

2. Background and Motivation

This section provides an overview of an existing context transfer (CT) protocol, and describes security contexts, applications thereof, motivation behind this work.

2.1. Context Transfer Protocol

CXTP [4] is a protocol in which security contexts are sent and received with the aim of quickly re-establishing security CT-candidate services, such as those of authentication, authorization, and accounting (AAA) registration keys for mobile IP [14,15], IP header compression [16,17], and AAA message exchange for the IEEE 802.1X [18,19], without requiring an MN to explicitly perform all protocol flows for these services from scratch. CXTP deals with two distinct scenarios.

In the first scenario, a previous access router (pAR) receives a CT trigger from the MN with the help of the IEEE 802.21 Standard that is responsible for handover negotiation and layer-2 connectivity, and then proactively routes a context-transfer data (CTD) message containing an involved security context to a next access router (nAR), referred to as 'predictive routing'. However, how to select the target nAR in a predictive manner is beyond the scope of the specification of CXTP—filling this gap is part of goals of this work.

In the second scenario, when the predictive routing in the first scenario fails, the nAR receives a CT trigger from the MN and then sends a context transfer request (CT-Req) message to the pAR. In response to the CT-Req message, the pAR routes a CTD message to the nAR. The nAR then replies with a CTD-reply (CTDR) message. The second scenario is referred to as 'on-demand routing'. We will elaborate on these two routing methods when presenting the design of our framework.

2.2. Security Contexts and Applications

Transfer-candidate services determine types of security contexts that they use. Although the concept of security contexts is not restricted, for the clarity purposes, two types of security contexts are specified in this paper. First, security contexts for mobile IP contain an AAA registration key required for the secure association [15]. When an MN undergoes a handover and attaches itself to an foreign agent (FA), it requests a registration key from its home AAA server *via* the FA. Upon receiving the registration key, the MN verifies a reply message from the FA, creating a security association with the FA [15]. However, the generation of the registration key requires message exchanges through the AAA infrastructure [20] (e.g., a foreign AAA server contacts the MN's home AAA server), causing a significant delay [17]. Such delays can be reduced greatly by routing a security context containing the registration key to the FA from the MN's home agent (HA) or previous FA.

Like the registration key for mobile IP, security contexts for the IEEE 802.1X Standard [18] contain the AAA key material. The IEEE 802.1X Standard provides port-based network access control for devices that attempt to attach themselves to a LAN port. The framework positioned behind an AP has three options in access control: (1) no authentication in which no filter is applied, (2) wired equivalent privacy (WEP) [21] which was originally designed to prevent wireless communication from eavesdropping, but found to have serious security weaknesses, and (3) open authentication—all packets are filtered out except for EAPOL (EAP over LAN) [18]—in which any third-party security mechanisms are applied, in conjunction with the IEEE 802.11i [22]. In particular, the 802.11i framework is specified jointly with the AAA infrastructure, in which the AP serves as an AAA client (or called an authenticator in the case of the IEEE 802.1X Standard).

Given the MN's master key shared with its AAA server, the MN is authenticated *via* the AAA client (AP) by exchanging messages with the AAA server (e.g., RADIUS [23] or diameter [24]), thereby resulting in a pairwise master key (PMK) of 256 bits for the AAA server, AAA client, and MN. The PMK is used

DOI: 10.1002/sec

to derive a pairwise temporary key (PTK), with the AAA client and MN communicating with each other. The PTK is a per-AP key that is associated with each specific AP, allowing for the protection of the wireless link between the AP and MN. Thus, the MN's re-association with another AP requires a new PTK, most likely by interacting with the AAA server. This interaction, however, incurs a significant delay which, in the case of 'roaming' or cross-handovers, is proportional to the round-trip time between the two involved AAA servers [25]. Thus, routing a security context containing the PMK(s) for authentication should reduce the authentication latency [26–28].

2.3. Motivation

A legacy scheme that provides secure cross-handovers for mobile users does not guarantee that the underlying services remain seamlessly operational, due to the nature of the Internet *via* which the involved AAA servers are connected. The delay in communications between the AAA servers varies with their link quality, and is also proportional to the round-trip time between the two servers. Figure 1 illustrates the procedure involved with the secure cross-handover. The MN's association with an AP triggers the initiation of an authentication protocol (e.g., diameter or MAP [28]), thereby generating a PTK *via* the IEEE 802.11i Standard (which corresponds to Step 1). When the MN crosses a domain boundary and associates with an AP in another domain (Step 2), the AAA foreign client (AAAFC) on the AP forwards the AAAF (AAA foreign server) a roaming security-context request, eventually reaching the AAAH (AAA home server) (Steps 3 and 4). The AAAH responds with a newly generated PMK to the AAAF (Step 5). As with the PTK generation mechanism, the MN and AP in the other domain possess a PTK to secure their link (Step 6).

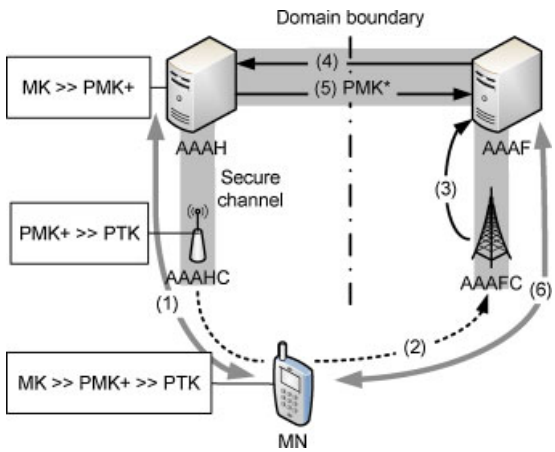

Fig. 1. An AAA-based legacy scheme for securing cross-domain handovers: AAAH denotes the MN's home AAA server that controls an AAAC (AAA client). Likewise, AAAF denotes a foreign AAA server that has a security association and administrative agreement with the AAAH.

For the seamless handover, the overall procedure should be completed before the link between the AAA home client (AAAHC) and MN is able to be disconnected, or the disconnection remains only within an acceptable threshold, e.g., 50 ms for VoIP [29]. The link between the AAAH and AAAF is the most dominating factor in causing the delay that varies with network traffic. Nevertheless, current systems lack the capability of 'pushing' the security context to the AAAF over an IP-network.

In this paper, we cope with such a pushing mechanism by which the security contexts are exchanged in a predictive manner. By doing so, we expect to avoid the re-establishment of involved security sessions from scratch while the MN is on the go. On the other hand, the decision to be made on the predictive routing of the security contexts is highly subject to the MN's mobility pattern. For instance, when most MNs are likely to associate with a group of specific APs, we may have only to route corresponding security contexts to the group. The selective, predictive routing of the security contexts that we want to offer will achieve secure and seamless cross-handovers.

3. MN's Mobility

This section presents a simple mobility model for evaluation purposes, based on which we simulate the MN's movement on a grid. According to some probability distribution that determines movement patterns, the mobility model typically selects two or more elements including speed, distance, angle, destination, or travel time. Note that the selection of these elements is not subject to a series of movements (of a single MN). We thus combine some of these elements and a probability distribution thereof, resulting in different mobility models [30].

3.1. Mobility Model

Our model selects speed (v) and angle (θ_t) given time, t. v is a random speed following a Gaussian distribution with a mean μ_v and variance σ_v, chosen from [0, 100 mph]. θ_t is also a random angle following a Gaussian distribution with a mean μ_a and variance σ_a,

DOI: 10.1002/sec

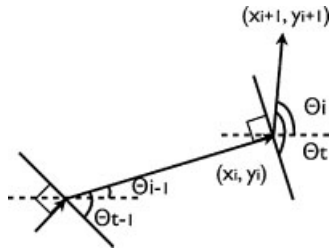

Fig. 2. Relative anchor-based directional changes.

chosen from $[0, \pi]$. In particular, as shown in Figure 2, θ_t is based on a *relative anchor*, a base line perpendicular to the previous direction. In this figure, we obtain θ_i by calculating $\theta_t + \theta_{i-1} - \frac{\pi}{2}$. Thus, given a monitoring interval, Δt, an MN's location is calculated as follows:

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} \cos\theta_i \cdot v \cdot \Delta t \\ \sin\theta_i \cdot v \cdot \Delta t \end{pmatrix}$$

This way, we mitigate abrupt changes in direction. Clearly, the people's movement with sharp turns can appear frequently, while such turns rarely do in the vehicles' movement. Cast studies of this kind are also found in References [9,31].

3.2. Assumptions Underlying Our Approach

We tend to track a sequence of the MN's association with APs while the MN undergoes handovers. First, APs assume to be randomly and uniformly distributed, covering the MN's movement paths. Second, APs' signal strength varies, resulting in the different size of their coverage. Under these two assumptions, Figure 3(b) shows the MN's association with APs, zooming in on a portion of its trace shown in Figure 3(a). At first, the MN keeps associated with ap_a until reaching position p_1 on the path. At p_1 where ap_b's signal strength is greater than ap_a's, the MN associates itself ap_b. The MN, then, becomes connected to ap_c at position p_2 for the same reason. Just before reaching position p_3, three signals from ap_c, ap_d, and ap_e are additionally detected, and since ap_e's signal strength at p_3 is greater than that of the others, the MN associates itself with ap_e. The cycle of disassociation and association continues based on the signal strength, resulting in a sequence of the MN's associations: ap_a, ap_b, ap_c, ap_e, ap_d, and ap_f. The sequence is translated into a time-varying sequence with the sojourn time specified at each AP, e.g., (ap_a, t_a).

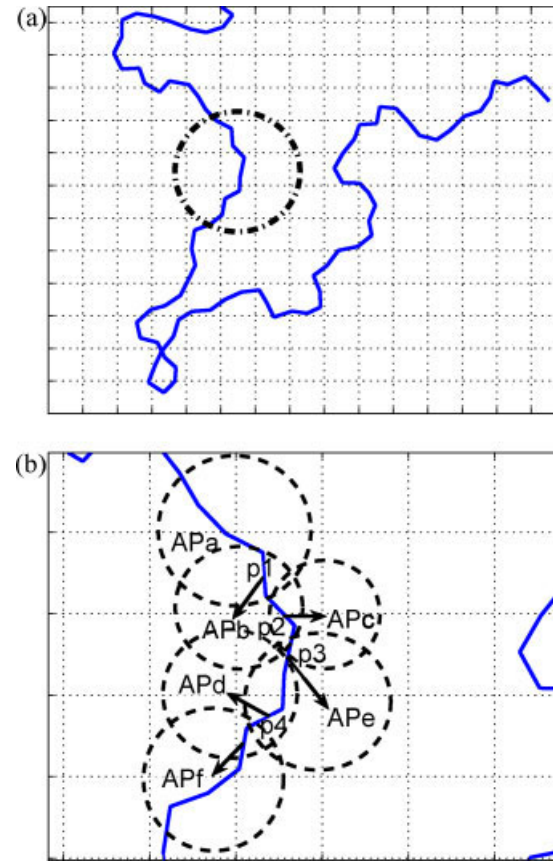

Fig. 3. An MN's movement trace and association with APs, with μ_a, σ_a, μ_v, σ_v each set to $\pi/2$, 1, 10 mph, and 1. (a) An MN's movement trace; (b) zoomed-in-on circle.

4. The Context Router

The CXR is responsible for monitoring the MNs' movement, detecting the likelihood of their cross-handover, analyzing the MN's movement patterns, and routing the corresponding security context to the selected CXR(s). This section describes three key components, i.e., monitoring, analysis, and routing.

4.1. Monitoring

Timely detection of the likelihood of an MN's cross-handover is of great importance to the predictive routing of security contexts. For the detection of the cross-handover, the MN issues a trigger message to the CXR *via* the IEEE 802.21 Standard. The MN's trigger message, however, does not usually include routing information on a target CXR, e.g., its identity and IP address. Thus, the target CXR needs to identify the MN, requiring its security context. When

DOI: 10.1002/sec

the information is not available, all relevant protocol flows are re-established from scratch. Alternatively, the binding CXR is aware of the cross-handover detection whenever the MN associates itself with an AP in the boundary. This can be realized by tracking the MN's movements.

The CXR tracks the MN's association with APs *via* an AP queue (APQ). The APQ is a list on which the MNs are registered when they are bound to the CXR. The CXR, cxr_j needs to verify whether the AP (with which the MN is associated) is on a domain boundary. For this, a boundary neighbor register (BNR) on cxr_j is defined as a triple $(\text{ap}_j, \text{cxr}_k, \text{ap}_k)$, where ap_j is an AP on the cxr_j's boundary, and ap_k is an AP on the cxr_k's boundary, adjacent to ap_j. The relationship between ap_j, cxr_k, and ap_k is established and registered with the cxr_j's BNR when the MN undergoes a cross-handover from ap_j to ap_k, leading cxr_k to add (ap_k, cxr_j, and ap_j) to its BNR. The relationship, however, does not necessarily mean that cxr_j and cxr_k are physically adjacent to each other; however, they are 'logical' neighbors to each other (i.e., one reaches the other through an n-hop route).

For the MN's directional pattern, a time-varying association sequence is represented by a TVA. The TVA is a sequence of changes in direction which are calculated with the associated APs' location. The TVA is eventually encoded into a sequence of *indices*. The sequence of indices represents the MN's movement pattern that will be used to predict its future cross-handover. The detail of creating the index sequence is as follows.

First, a time-varying association sequence $[(\text{ap}_a, 4), (\text{ap}_b, 5), (\text{ap}_c, 5), (\text{ap}_e, 4), (\text{ap}_d, 6), (\text{ap}_f, 4)]$, where the second elements represent time steps as a sojourn time at each AP indicated by the first elements, is recorded. In particular, the association sequence can be shrunk by adjusting the time interval at which the association is stamped on the sequence. For instance, when the time interval is set to three steps, the sequence appears as $[\text{ap}_a, \text{ap}_b, \text{ap}_c, \text{ap}_e, \text{ap}_d, \text{ap}_d, \text{ap}_f]$ in the case where each AP monitors, or $[\text{ap}_a, \text{ap}_b, \text{ap}_b, \text{ap}_c, \text{ap}_e, \text{ap}_e, \text{ap}_d, \text{ap}_d, \text{ap}_f]$ in the case where the CXR monitors; the former is more representative for changes in direction than the latter—a similar approach is also found in Reference [32]. Once the association sequence is provided, the slope from a pair of ap_a and ap_b, i.e., θ_{ab} with their coordinate is calculated. This way, the MN's movement is abstracted. Likewise, θ_{bc} is calculated with ap_b's and ap_c's coordinates. This calculation repeats until the last with a pair of ap_d and ap_f completes. Thus, the TVA having $[\theta_{ij}, \theta_{jk}, \cdots, \theta_{mn}]$ is generated. Next, the TVA is *encoded* by quantizing angles:

$$\frac{2\pi}{n}(k-1) \leq \theta < \frac{2\pi}{n}k, \quad k \leq n \in \mathbb{Z} \tag{1}$$

where n is the total number of quantized angles. When $n = 8$, they lead in eight directions, which we will keep in the remainder of this paper. Note, however, that this model can be generalized with the number of directions adjusted. As a result from the encoding, each direction is labeled with an index of the quantized angle, k, and then the corresponding TVA is eventually transformed into an index sequence $[i, j, \cdots, m]$, where i, j, and $m \leq 8$.

4.2. Analysis

An index sequence as a new pattern is added to a pattern data register (PDR). The PDR is a set of entities, each of which consists of a unique pattern that is represented by an index sequence and distribution. The distribution, *pd*, is created with pairs of a neighbor CXR and degree to which the index sequence is statistically bound to each neighbor CXR. That is, as the MN having a specific pattern is tightly bound to a specific CXR, the degree to the CXR is increased. *pd* is denoted by

$$\text{pd} = \{(\text{cxr}_1, \text{dg}_1), \cdots, (\text{cxr}_n, \text{dg}_n)\} \tag{2}$$

Using this distribution, we select neighbor CXRs to which the MN will be bound with a certain probability. The selection of the CXRs involves two main steps: matching patterns and filtering out the CXRs having a low degree.

4.2.1. Matching Patterns

Matching patterns is to gauge (dis)similarities between an observed TVA and each *a priori* pattern stored in the PDR. We represent the similarity with the distance that is calculated by *editing* one pattern to make it the same as the other (i.e., edit distance [12]), or alternatively applying the χ^2-distance [12]. The advantage of the two methods is to provide a threshold which we adjust for the degree of the similarity as follows:

$$f(a, b) < \delta_d \tag{3}$$

where a and b are pattern sequences, and f is either the editing function, e, or χ^2-distance calculating function. The smaller the value of δ_d, the more fine-grained the matching can be, and the more specific the group in

DOI: 10.1002/sec

which the TVAs are accepted as the same. The number of pattern sequences in the PDR, however, can grow, likely increasing the computation complexity. To contain the complexity, a multi-class support vector machines (SVM) technique [33–35] can be applied. In SVM, all pattern sequences are mapped into a multi-dimentional feature space in which the pattern sequences are transformed and then used to generate *support vectors*. These support vectors form the (multiple) hyperplane(s) by which all the sequences are classified. If this is the case, only support vectors are stored and compared with new pattern sequences.

Function e takes two indexed TVAs: $a_1 \cdots a_n$ and $b_1 \cdots b_n$, forming an $(n+1) \times (n+1)$ matrix. The matrix is set as follows: $e(1,1) = 0$, $e(1, b_1 \cdots b_n) = 0$, and

$$e(a_1 \cdots a_i, 1) = \begin{cases} 1, & i = 1 \\ \infty, & 1 < i \leq n \end{cases}$$

Then, to make a the same as b, we replace or delete a sample in a, or insert the same as a sample in b, repeatedly, until reaching the last sample in a. Function e is given by

$$e(a_i, b_j) = \begin{cases} e(a_{i-1}, b_{j-1}) & \text{if } a_i = b_j \\ \min \begin{pmatrix} e(a_{i-1}, b_{j-1}) + C_r, \\ e(a_{i-1}, b_j) + C_d, \\ e(a_i, b_{j-1}) + C_i \end{pmatrix} & \text{if } a_i \neq b_j \end{cases} \tag{4}$$

where C_r, C_d, and C_i are replacement, deletion, and insertion costs, respectively. They are set to 1 in our evaluation.

Alternatively, χ^2-distance-based techniques are found in diverse areas such as scene-change detection in image sequences [36,37] and anomaly detection [38]. The calculation of the χ^2-distance is given by

$$\chi^2(a, b) = \sum_{i=1}^{n} \frac{(a_i - b_i)^2}{(a_i + b_i)} \tag{5}$$

Clearly, $\chi^2 = 0$ if and only if all samples in a match those in b. The higher the value of χ^2, the less likely the observed TVA fits the expected pattern. The χ^2-distance function is computationally less expensive than the editing function, which will be evaluated in detail in Section 5.

4.2.2. Filtering

When an observed TVA matches a pattern in the PDR, the CXR fetches the corresponding distribution (see Equation (2)). It then applies either of two filtering methods to select neighbor CXRs. The first method is a filter based on Chebyshev inequality [39]. Given a normalized *pd* with a random variable X with a finite mean μ and finite standard deviation σ, the Chebyshev inequality provides a relationship between σ and $|X - \mu|$ such that

$$P\left(|X - \mu| \geq \frac{\sigma}{\sqrt{\delta_p}}\right) \leq \delta_p \tag{6}$$

The inequality relationship holds for any dropout rate $\delta_p > 0$. This relationship-based filter is effective when *pd* is a normal distribution [39], reflecting a strong relationship between a given pattern and its association with a neighbor CXR.

The second method is a cutoff-based filter using a threshold, δ_r, on a scale of 0–1 by which top $n\%$ neighbor CXRs from the normalized *pd* are selected. In particular, when $\delta_r = 0$, no CXRs are selected, and hence only the on-demand routing, which will be described shortly, remains effective. When $\delta_r = 1$, all neighbor CXRs are flooded with security contexts. When $0 < \delta_r < 1$, security contexts are routed to the selected neighbor CXRs.

4.3. Routing

After the filtering, (the replicas of) security contexts are routed to the selected neighbor CXRs either in a predictive manner or on-demand. The security contexts are stored in a context register (CR) on the corresponding neighbor CXRs. If the MN is bound to one of the selected neighbor CXRs (nCXR) in which the MN's security context is immediately available, then, it will not experience any extra delay in fetching its security context from the previous CXR (pCXR) (i.e., predictive routing). By contrast, the MN may be eventually bound to an unexpected neighbor CXR (uCXR), i.e., a security-context miss. When the security-context miss occurs, the CXR to which the MN is currently bound fetches its security context from the pCXR (i.e., on-demand routing). After the security context is routed either in a predictive manner or on-demand, the association between pCXR and uCXR, and between their APs are added to, or updated on, both pCXR's and uCXR's BNRs. The same holds for the corresponding distribution (Equation (2)) in their PDR.

DOI: 10.1002/sec

The growth of security context replicas varies, depending on the MNs' location on the domain boundary and their cross-handover patterns. In particular, some MNs are likely to cross the boundary and eventually stay thereon or cross back while some MNs may cross the boundary back and forth in a short time period, i.e., a *ping-pong effect*. This effect causes a rapid increase in the number of replicas on the corresponding CXRs. For this reason, to limit their excessive growth, we apply two rules. The first rule is to fix the buffer size for replicas. When the buffer is full, the replicas in the buffer are invalidated in the least recently used order—the invalidation is less costly than the deletion—by simply toggling off the validity bit. This rule is effective against the rapid growth of replicas. The second rule is to limit the time in keeping the replicas in the CR. That is, incoming replicas are time-stamped with a timer threshold. When the timer expires, the replicas in the CR are scanned in comparison with the threshold, timeout, and time-stamped time, eventually invalidating stale replicas. This rule has a tradeoff between storage savings and computation performance.

4.4. Prototypical Design

Figure 4 illustrates the software implementation of the CXR and its interfaces with another CXR *via* CXTP [4] and with MNs *via* the IEEE 802.21 Standard.

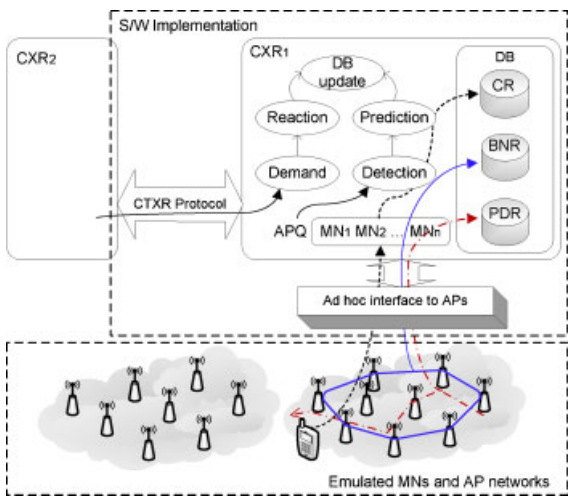

Fig. 4. The CXR software implementation. An access network is formed by APs whose location is represented by an xy-coordinate on a grid. An individual MN moves over the grid based on its own mobility model with the values of system parameters changed.

Predictive routing:
Input: δ_p: a probability threshold
Input: $eProb$: an estimated crossover probability
δ_c: a cutoff threshold
Input: pd: the probability distribution from the BNR
Input: $Context$: a security context from the CR

If $eProb > \delta_c$ **then**
Perform either CutOffSelection(δ_c, pd) or
StatProbSelection(pd), returning $CXRs$
Route $Context$ to $CXRs$
end if

Fig. 5. Pseudo code: decision on the predictive routing.

The APQ on the CXR is a registration list on which the MN currently bound to the domain is recorded along with its sojourn time and association with APs; using this information, the MN's movement is tracked. Given a time interval, the APQ is scanned to find whether any MN is associated with APs inside the boundary in reference to the BNR. If this is the case, the detection state is transitioned to the prediction state. In the prediction state, the CXR performs the predictive routing of the security context according to the cross-handover probability (e.g., 7 times of success in 10 trials of predictive routing results in the probability of 0.7). So, if the cross-handover probability is greater than the threshold, the CXR applies the prediction algorithm—its pseudo code is given in Figure 5, thus routing the security context. The demand state, on the other hand, is initiated by a request from other CXRs (e.g., by a CT-Req message in CXTP), ultimately transitioning to the reaction state in which the on-demand routing is performed.

5. Evaluation

Success in the predictive routing of security contexts implies substantial reduction in handover delays. Such delays, in particular, involving message exchanges by the underlying applications, vary greatly, depending on the round-trip time between two communication entities. For instance, in the case where two AAA servers are involved for secure communications, the average delay in establishing a security association is proportional to the round-trip time between the two servers [25]. For this reason, our evaluation focuses on prediction accuracy. At the same time, since many MNs cross the domain boundary, computation performance for the prediction mechanisms is also an important factor in the evaluation. Thus, the metrics used to indicate prediction and computation performance include accuracy and CPU utilization. Accuracy is represented by $(TP + TN)/(TP + TN + FP + FN)$, where TP (true positive)—the number of times the selection

DOI: 10.1002/sec

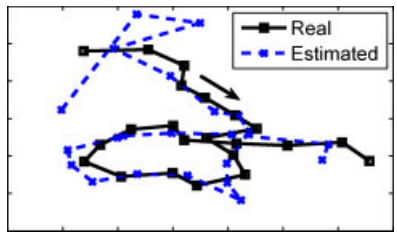

Fig. 6. Movement path estimation with the Kalman filter.

of a neighbor CXR is correctly estimated; TN (true negative)—an MN is estimated to stay on its current CXR; FP (false positive)—an MN is bound to an unpredicted CXR; and FN (false negative)—an MN leaves its current CXR although it was predicted to stay thereon.

5.1. Kalman Filter

We evaluate prediction accuracy in comparison with the Kalman filter [13]. In applying the Kalman filer, a state is specified by the MN's location (x and y) and speed ($\mathrm{d}x/\mathrm{d}t$, $\mathrm{d}y/\mathrm{d}t$), and we set values in process and measurement noise covariances, Q and R, to 0.1 and 1, respectively. In addition to these parameters, we set values in transition matrix A and measurement matrix H as

$$\begin{pmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Figure 6 shows the efficacy of the Kalman filter; estimation accuracy is determined by two factors: measurement and past states. At the time of initialization, however, the past state is unknown, and hence the Kalman filter performs in an ad hoc manner. Even with a poor initialization, however, it quickly converges to the true values.

In the following, we first describe the simulation of MNs and access networks. Then, after assessing system parameters defined in our prediction mechanisms, prediction accuracy is evaluated with the local optimal values of the obtained system parameters. Finally, we analyze computation performance for the prediction mechanisms.

5.2. Simulation of MNs and Access Networks

APs are assumed to be randomly and uniformly distributed, forming a virtual access network. We have chosen a grid with a tradeoff between practice and simplicity. On a grid, each node has eight neighbors except for the nodes positioned at the edge of the grid. Its network size is a 20×20 grid, which is partitioned into 25 domains (i.e., CXRs). The grid allows for ease of simulating and tracking the MN traffic.

Each MN moves, according to its own mobility model while being associated with different APs. At a given time t, the CXR obtains the MN's location and then calculates the location of the AP closest to the MN by applying a 2-level hash function such that APs on the grid are filtered, based on the MN's x coordinate. This results in an array of APs closest to the x coordinate. The array is again filtered based on the MN's y coordinate, thereby obtaining the location of the AP closest to the MN. This way, the CXR calculates the location of the MN and corresponding AP. The calculation recurs at $t + \Delta t$, where Δt is the monitoring interval. Δt is given, based on the MN's maximum speed, $V_{\max}$, i.e., $\Delta t = d/V_{\max}$, where d is the distance between the MN's current and previous APs. Obviously, d is smaller than the distance (D) between the centers of the two involved APs, assuming that the two APs' coverage areas (circles) overlap to the extent that $d = D/2$. Δt is adjusted to capture a series of the MN's association with APs. The smaller the value of Δt, the more fine-grained the measures, but the higher the overhead thereby.

Figure 7 shows MNs' movement traces and their association with APs. The smaller the value of Δt, the more fine-grained traces we can obtain, in comparison with Figure 7(a) and (b). Obviously, the monitoring frequency governs a tradeoff between the accuracy in prediction and the overhead for the monitoring. The monitoring frequency is bound to a certain degree of accuracy due to the inherent uncertainty/noise and abstraction on the traces. For instance, Figure 7(c) and (d) each are the abstracted traces of Figure 7(a) and (b). Figure 7(c) is coarse-grained, compared to Figure 7(d), resulting in low accuracy in prediction. By contrast, when the MNs' movement pattern is considered *ad hoc*, the coarse-grained abstraction can be sufficiently effective.

5.3. Assessment of System Parameters

System parameters considered for the evaluation include the TVA length combined with the matching threshold δ_d, cutoff threshold δ_r, and dropout rate δ_p. In order to assess these parameters, we create 100 MNs that move based on their mobility model and observe their movement patterns; each MN undergoes handover 1000 times. First, when the TVA length increases from 6 to 15 with δ_d set so as to let two TVAs match more than

DOI: 10.1002/sec

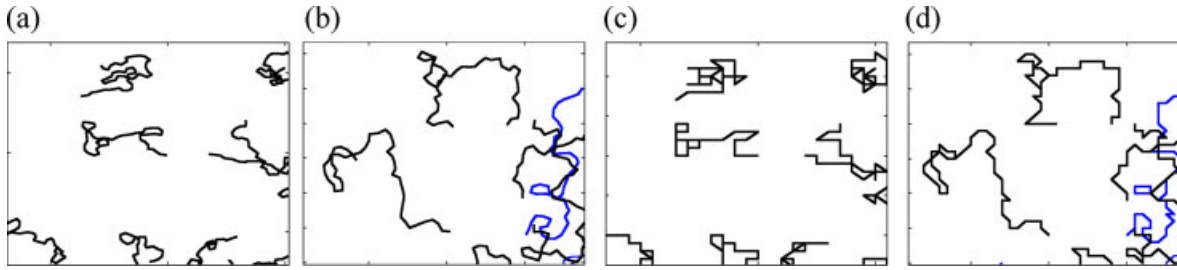

Fig. 7. MN trace and abstraction. (a) MN traces in Δt; (b) MN traces in $\Delta t' = \Delta t/2$; (c) AP-association traces of (a); and (d) AP-association traces of (b).

70% of times, the 9- and 13-length TVAs yield locally highest prediction accuracy. When δ_d is set to let 60% of two TVAs match, the 11-length TVA results in the highest prediction accuracy. Thus, both the TVA length and its threshold must be tuned together for the best results, based on the edit distance technique applied. We set the TVA length and δ_d to 9 and 70%, respectively, throughout the evaluation. In addition to these parameters, we set each δ_r and δ_p to 5 and 0.5 which appear almost constant.

For the ping-pong effect, depending on the cross-handover probability, the threshold is adjusted. When the MN is unlikely to cross back immediately, the predictive routing may not be applied just after the MN's cross-handover. For this reason, it is important to find an optimal cross-handover probability threshold that is tuned to the MNs' movement patterns. Figure 8 shows the relationship between the threshold and false alarm rate including the false negative and false positive rates. To some degree, the lower the threshold, the more aggressive and frequent the predictive routing. When the threshold decreases below the value of 0.3, the false-alarm rate goes up to 0.44, implying more MNs that have crossed the boundary are likely to stay thereon although they are predicted to cross back. When the threshold increases above the value of 0.4, the false-alarm rate also goes up to 0.49, implying that more MNs are likely to cross back although they are predicted to stay thereon. Accordingly, the cross-handover probability threshold ranging from 0.3 to 0.4 is set to cope with the ping-pong effect.

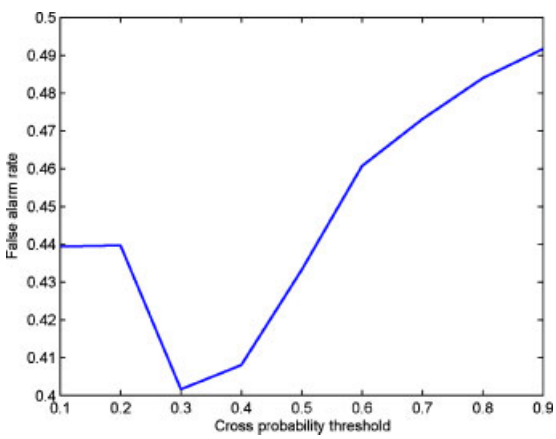

Fig. 8. False alarm *versus* cross-handover probability threshold.

5.4. Evaluation Results

Four prediction mechanisms have been described in Section 4. The cutoff-based and inequality relationship-based mechanisms are combined with the edit-distance and χ^2-distance functions. Prediction accuracy is related to not only the similarity measurement techniques, but also the classification of patterns into groups. Each group is represented by the probability distribution *pd* on which the prediction mechanisms depend. In general, the inequality-based mechanisms are effective for *pd* that is normally distributed with even a large variance, while the cutoff-based mechanisms operate well with various distributions. As shown in Figure 9, prediction accuracy offered by the inequality-based mechanisms almost equals that by the cutoff-based ones. In which case, selecting a similarity measurement technique determines the performance of prediction accuracy. That is, applying the edit distance technique achieves up to 1.4 times higher prediction accuracy than applying χ^2-distance.

Since the Kalman filter yields an estimated future location of the MN, we select the AP closest to the estimated location and then the neighbor CXR that controls the AP. Note that the selected AP may not be on the domain boundary, which is not found in the BNR. This case requires additional information on the binding of a neighbor CXR and its APs to be available to another CXR. Conversely, our prediction mechanisms capture an AP association pattern based on the MN's movements, limiting the

DOI: 10.1002/sec

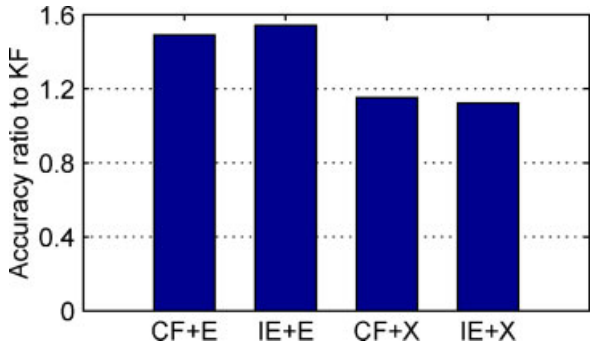

Fig. 9. Average prediction accuracy comparison of the four mechanisms with the Kalman filtering (KF): CF and IE stand for cutoff and inequality, and E and X for edit and χ^2 distances, respectively.

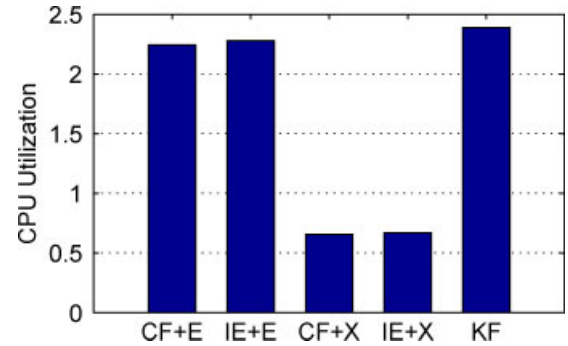

Fig. 10. Average CPU utilization comparison.

number of neighbor CXRs to which to route the corresponding security context. Thus, for the fairness purposes, the cutoff- and inequality-based mechanisms *select one neighbor CXR*. Also, the Kalman filter-based estimation is deferred until the BNR is populated, and the same holds for our mechanisms.

As shown in Figure 9, all the four prediction mechanisms outperform the Kalman filter-based approach. Particularly, applying the edit distance to the inequality- and cutoff-based mechanisms causes them to outperform the Kalman filter-based approach by 54 and 49%, respectively. Similarly, applying the χ^2 distance to those mechanisms achieves up to 15% higher accuracy. Meanwhile, we also measured prediction accuracy of the flooding method of routing security contexts to all neighbor CXRs (i.e., $\delta_r = 1$). The flooding method is only 6% more accurate than the inequality-based mechanism with the edit distance applied, which underachieves in the sense that the other mechanisms pinpoint a single CXR. Rather, our mechanisms impose less heavy network traffic than the flooding method—the traffic created by the flooding method is proportional to the number of neighbor CXRs. This advantage grows as the number of MNs likely to undergo a cross-handover increases. Considering that tens of millions MNs move round, this advantage must be of great benefit.

Based on the accuracy data shown in Figure 9, we can project how much benefit we can get *via* the predictive routing based on the best of the four, i.e., the inequality-based mechanism having the edit distance. Statistically speaking, the predictive routing causes the cross-handover to be on average 2.5 times faster than the reactive routing that is required to fetch security contexts. This improvement linearly increases as the delay in fetching grows.

When it comes to computation performance, applying the χ^2-distance is three times more efficient than applying the edit-distance, since the edit-distance requires to perform many comparison operations at the expense of prediction accuracy, as shown in Figure 10. Similarly, applying the χ^2-distance to the inequality-based mechanism is 3.6 times more efficient than the Kalman filtering. Accordingly, the inequality-based mechanism with the χ^2-distance applied is the best option for computation performance, while applying the edit-distance to either the cutoff- or inequality-based mechanism is the one for prediction accuracy.

6. Conclusion

As MNs undergo cross-handovers, their security contexts must be effectively and efficiently transferred and managed both on-demand and in a predictive manner. This paper presented a CXR, an integrated framework for security-context management in the cross-handover, with the aim of furthering users' mobility and seamless secure execution of their applications on their mobile devices. We began by characterizing the MN's movement patterns and then designed the CXR with three important steps. We analyzed prediction accuracy in routing (the replicas of) the security contexts ahead of the MN's arrival. The evaluation result shows that the predictive routing causes cross-handovers to be 2.5 times faster than the reactive routing. In summary, we have (1) provided an integrated framework in combination of the security context routing methods and prediction mechanisms with the patterns extracted from the association with APs, and (2) achieved the efficacy, deployability, and scalability of security context management. As a consequence, our CXR allows for seamless and secure services with predictive routing of security contexts enabled.

DOI: 10.1002/sec

References

1. Kim H, Shin KG. Predictive routing of contexts in an overlay network. In *IM*, IFIP/IEEE, June 2009.
2. Gupta V, Das S, Cypher D. http://www.ieee802.org/21/
3. Koodli R, Perkins CE. Fast handovers and context transfers in mobile networks. *SIGCOMM Computer Communication Review* 2001; **31**(5): 37–47.
4. Loughney J, Nakhjiri M, Perkins C, Koodli R. Context transfer protocol. *RFC 4067*, July 2005.
5. Mishra A, Shin M, Arbaugh W. Context caching using neighbor graphs for fast handoffs in a wireless network. In *INFOCOM*, Hong Kong, IEEE, March 2004.
6. Chou C-T, Shin KG. Smooth handoff with enhanced packet buffering-and-forwarding in wireless/mobile networks. *Wireless Networks* 2007; **13**(3): 285–297.
7. Song L, Deshpande U, Kozat U, Kotz D, Jain R. Predictability of wlan mobility and its effects on bandwith provisioning. In *INFOCOM*, Barcelona, Spain, IEEE, April 2006.
8. Akyildiz I, Wang W. The predictive user mobility profile framework for wireless multimedia networks. *Transactions on Networking* 2004; **12**(6): 1021–1035.
9. Liu T, Bahl P, Chlamtac I. Mobility modeling, location tracking, and trajectory prediction in wireless atm networks. *Journal on Selected Areas in Communications* 1998; **16**(6): 922–936.
10. Soh W-S, Kim HS. Dynamic bandwidth reservation in cellular networks using road topology based mobility predictions. In *INFOCOM*, Hong Kong, IEEE, March 2004.
11. Liang B, Haas Z. Predictive distance-based mobility management for pcs networks. *Transactions on Networking* 2003; **11**(5): 718–732.
12. Duda RO, Hart PE, Stork DG. *Pattern Classification* (2nd edn). Wiley-Interscience: New York, 2001. ISBN 0-471-05669-3
13. Welch G, Bishop G. An introduction to the kalman filter. *Technical Report TR 95-041*, UNC at Chapel Hill, July 2006.
14. Allard F, Bonnin J-M. An application of the context transfer protocol; ipsec in a ipv6 mobility environment. *International Journal of Communication Networks and Distributed Systems* 2008; **1**(1): 110–126.
15. Perkins C, Calhoun P, Burmeister C, Degermark M. AAA registration keys for mobile IP. *IETF Draft*, work in progress, 2003.
16. Bormann C, *et al*. Robust header compression (rohc): framework and four profiles: RTP, UDP, ESP, and uncompressed. *RFC 3095*, July 2001.
17. Westphal C, Koodli R. IP header compression: a study of context establishment. In *WCNC*, IEEE, March 2003; 1025–1031.
18. IEEE Computer Society LAN MAN Standards Committee. *Technical Report ANSI/IEEE* Std 802.1X-2001, IEEE, 2001.
19. Mishra A, Shin M, Arbaugh W. Pro-active key distribution using neighbor graphs. *Wireless Communications Magazine* 2004; **11**(1): 26–36.
20. Authentication authorization and accounting IETF WG. http://www.ietf.org/html.charters/aaa-charter.html
21. WEP algorithm. http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html
22. Aboba B, *et al*. Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: specification for robust security. *Technical Report IEEE* Std 802.11i/D3.1, IEEE, 2003.
23. Rigney C, Willens S, Rubens AC, Simpson WA. Remote authentication dial in user service. *RFC 2865*, June 2000.
24. Calhoun PR, Loughney J, Guttman E, Zorn G, Arkko J. Diameter base protocol. *RFC 3588*, September 2003.
25. Kim H, Afifi H. Improving mobile authentication with new AAA protocols. In *ICC*, IEEE, Anchorage, USA, May 2003; 497–501.
26. Georgiades M, Akhtar N, Politis C, Tafazolli R. Enhancing mobility management protocols to minimise aaa impact on handoff performance. *Computer Communications* 2007; **30**(3): 608–618.
27. Shin M, Ma J, Mishra A, Arbaugh WA. Wireless network security and interworking. In *Proceedings of the IEEE* 2006; **94**(2): 455–466.
28. Kim H, Shin KG, Dabbous W. Improving cross-domain authentication over wireless local area networks. In *SecureComm*, IEEE, Athens, Greece, September 2005; 127–138.
29. Shirdokar R, Kabara J, Krishnamurthy P. A QoS-based indoor wireless data network design for VoIP. In *Vehicular Technology Conference*, vol. 4. IEEE, October 2001.
30. Yoon J, Liu M, Noble B. Sound mobility models. In *International Conference on Mobile Computing and Networking* , ACM, San Diego, California, September 2003; 205–216.
31. Kim M, Kotz D, Kim S. Extracting a mobility model from real user traces. In *INFOCOM*, IEEE, Barcelona, April 2006.
32. Choi S, Shin KG. Predictive and adaptive bandwidth reservation for hand-offs in QoS-sensitive cellular networks. In *SIGCOMM Computer Communication Review* , ACM, Vancouver, British Columbia, September 1998; 155–166.
33. Cristianini N, Shawe-Taylor J. *An Introduction to Support Vector Machines and Other Kernel-based learning Methods*. Cambridge University Press: New York, 2000. ISBN 0-521-78019-5
34. Keerth S, Lin C-J. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation* 2003; **15**(7): 1667–1689.
35. Chang C, Lin C. Libsvm: a library for support vector machines. http://www.csie.ntu.edu.tw/∼cjlin/libsvm, 2001.
36. Patel NV, Sethi IK. Compressed video processing for cut detection. *Vision, Image and Signal Processing* 1996; **143**(5): 315–323.
37. Ford RM, Robson C, Temple D, Gerlach M. Metrics for scene change detection in digital video sequences. In *IEEE International Conference on Multimedia Computing and Systems*, IEEE, Los Alamitos, CA, USA, 1997; 610–611.
38. Ye N, Chen Q. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality and Reliability Engineering International* 2001; **17**(2): 105–112.
39. Papoulis A, Pillai SU. *Probability, Random Variables and Stochastic Processes* (4nd edn). McGraw-Hill: New York, 2002. ISBN 0-07-366011-6

DOI: 10.1002/sec