

An authentication architecture for cross-domain mobility

Hahnsang Kim

▶ **To cite this version:**

Hahnsang Kim. An authentication architecture for cross-domain mobility. Networking and Internet Architecture [cs.NI]. Institut National des Télécommunications d'Evry, 2006. English. NNT : . tel-00408687

HAL Id: tel-00408687

https://theses.hal.science/tel-00408687

Submitted on 31 Jul 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.





Thèse de doctorat de l'Institut National des Télécommunications dans le cadre de l'école doctorale SITEVRY en co-accréditation avec l'Université d'Evry-Val d'Essonne
Équipe d'accueil : Planète, INRIA Sophia Antipolis

Spécialité

INFORMATIQUE

Par
Hahnsang KIM

Thèse présentée pour l'obtention du
Doctorat de l'Institut National des Télécommunications

AN AUTHENTICATION ARCHITECTURE FOR CROSS-DOMAIN MOBILITY

Soutenue le 28 Avril 2006 devant le jury composé de :

Director de thèse :	Hossam AFIFI	Professeur, INT d'Evry
Rapporteurs :	Refik MOLVA	Professeur, Eurécom
	Isabelle CHRISMENT	Maître de Conférence, Univ. Henri Poincaré
Examinateurs :	Kaisa NYBERG	Professeur, Helsinki Univ. of Tech. Finland
	Pascal URIEN	Professeur, ENST
	José ARAUJO	Chercheur senior, Alcatel
Membres invité :	Kang G. SHIN	Professeur, Univ. of Michigan USA

Thèse n° 06INT001

To my family

Acknowledgements

First of all, I would like to respectfully express my sincere gratitude to Dr. Isabelle CHRISMENT, Maître de Conférence at the Univercité Henri Poincaré and Dr. Refik MOLVA, Professor Class 1 at the Institut Eurécom for being the *rapporteurs* of my thesis.

I wish to thank Dr. Kaisa NYBERG, Professor in the Department of CSE at Helsinki University of Technology, Finland; Dr. Pascal URIEN, Professor at Group des Ecoles des Télécommunications-Telecom Paris (GET-ENST); and Dr. José ARAUJO, Research project manager in the French Research and Innovation Centre of Alcatel, for being interested in my work and for joining the committee examining my thesis.

I am deeply indebted to Dr. Kang G. SHIN, Professor in the Department of CS and the Founding Director of the Real-Time Computing Laboratory at the University of Michigan, U.S.A., for his guidance and invaluable comments.

Profound gratitude is due to Dr. Hossam AFIFI, Professor at the Institut National des Télécommunications-Evry Université (INT-Evry) who supervised my thesis. His constant encouragement enabled me to successfully complete it.

I will forever appreciate the efforts of Dr. Walid DABBOUS, Directeur de Recherche at INRIA who made available all the means necessary for my work in the projet Planète while advising me.

I am extremely fortunate to have worked with Dr. Thierry TURLETTI, Charge de

Recherche at INRIA who inspired me with his encouraging research efforts in the initial stages of my work at INRIA.

I am grateful to all the members of the projet Planète at INRIA, including Dr. Thierry PARMENTELAT, for their social and technical assistance; Dr. Robert DE SIMONE, Directeur de Recherche at INRIA for his genial interest in my work; Dr. Emmanuel LÉTY, Research engineer at UDcast for his considerate comments; and Dr. Jin-Young CHOI, Professor at Korea University for his endless support and help.

Finally, this thesis is dedicated to my mother whose constant, perpetual love is always in my heart and to whom I wish to extend my heartfelt thanks.

Résume

L'évolution des technologies sans fil favorise de plus en plus la mobilité, mais il reste cependant des problèmes non résolus liés à la sécurité. Le changement de contexte d'un réseau à l'autre nécessite des procédures qui infligent du délai lié aux procédures 'réseau' ainsi qu'au changement de lien. La sécurité étant requise dans les deux changements. La procédure d'authentification à un grand impact sur le délai de l'établissement du lien. Cela est encore plus tangible pour une authentification à travers plusieurs domaines. Il en sort que la conception de systèmes d'authentification rapides qui ne compromettent pas le niveau de sécurité est un grand challenge. Dans cette thèse, je propose de manière générale une solution permettant la minimisation de cette latence. L'architecture est décomposée en trois contributions. La première partie se focalise sur une architecture d'authentification décentralisée. Nous introduisons la notion de 'proxy' qui permet de réduire les couts en terme de délai entre points d'accès. Le résultat est un mécanisme permettant de trouver l'endroit de placement optimal dans une architecture arborescente pour minimiser le nombre de messages. Dans la seconde partie, nous proposons un protocole d'estimation de la mobilité (MAP) et de préparation au changement. Cette partie est essentiellement dédiée à des mobilité inter-domaine, coopérant avec les agents de député. La troisième partie traite d'un mécanisme dans l'accès qui permet à des routeurs de bordure de préparer le contexte avant le changement afin encore une fois de réduire les latences. On se base sur des modèles

théoriques de prédiction basés sur des arbres de prédiction et des estimations statistiques des trajets.

Abstract

The rapid growth of wireless device technologies is enabling seamless mobility, but there are still major concerns related to the performance of security. The handoff performance correlates with inter-wireless link switch latency and network layer latency, with security being required at both levels. Authentication latency has a significant impact, especially on the link switch phase, in the case of cross-domain mobility because of the requirement of remote contact with a home authentication server. Providing a solution to minimize the latency impact without degrading the level of security is a major challenge. In this thesis, we propose a high-performance authentication architecture to tackle the latency problem in fast inter-domain handoffs. The architecture consists of three contributions. First, we present a decentralized authentication scheme by introducing a 'deputy' agent in control of a group of access points. The collaborating deputy agents considerably reduce long-distance traffic of authentication messages. Then, we propose a mobility-adjusted authentication protocol (MAP) dedicated to cross-domain handoffs, cooperating with the deputy agents. The protocol leverages the concept of 'security context' to achieve minimum handshakes so that one can significantly reduce the authentication latency. Finally, we design a security context router (SCR) that extends the deputy agent to manage security contexts. The SCR realizes seamless cross-domain mobility with the predictive forwarding of security context that is characterized by approximate pattern matching and statistical estimation. The

contributions made by this thesis have transparently led to significant improvements in the performance of handoff processes without compromising high-level security.

Contents

DEDICATION iii

ACKNOWLEDGEMENTS iv

RÉSUME vii

ABSTRACT ix

1 Introduction 1

1.1 Motivation 1

1.2 Overview of Mobile Wireless Security 3

1.3 Requirements 4

1.4 Defined Entities 6

1.5 What We Achieved 7

1.6 Main Contributions 8

1.7 Thesis Outline 10

2 The State of the Art 13

2.1 The IEEE 802.11i Standard 13

2.1.1 The IEEE 802.11 Association 14

2.1.2 The IEEE 802.1X . . . 15
2.1.3 Authentication Authorization Accounting . . . 16
2.1.4 EAP-based Authentication Protocols . . . 18
2.1.5 4-Way Handshake . . . 19
2.1.6 Group-key Handshake . . . 23
2.1.7 Fast Authentication . . . 23
2.2 Authentication in 3GPP . . . 25
2.3 The Integration of WLAN and GPRS . . . 26

3 Decentralized Authentication Architecture 31
3.1 Introduction . . . 31
3.2 AAA-UMTS Application . . . 32
3.2.1 Evaluation of the AAA-UMTS Application . . . 35
3.3 Decentralized Authentication Scheme . . . 38
3.3.1 Mobile Authentication . . . 39
3.3.2 An Abstract Model . . . 40
3.4 Performance Evaluation . . . 43
3.5 Related Work . . . 45
3.6 Conclusions . . . 46

4 Mobility-adjusted Authentication Protocol 47
4.1 Introduction . . . 47
4.2 Overview of Authentication Mechanism and Requirements . . . 49
4.2.1 The IEEE 802.11i Authentication . . . 49
4.2.2 Cross-domain-related Protocols . . . 50
4.2.3 Design Requirements . . . 52

4.2.4 BAN Logic 54
4.3 MAP 56
4.3.1 Architecture 57
4.3.2 Communication between SCRs 58
4.3.3 Authentication 59
4.3.4 Defined Keys 59
4.3.5 Defined Messages 61
4.3.6 Message Exchanges 61
4.4 Security Considerations 68
4.4.1 Protocol Analysis 68
4.4.2 Possible Attacks 71
4.5 Performance Evaluation 73
4.5.1 Simulation Methodology 73
4.5.2 The Simulation Model 74
4.5.3 The Simulation Results 76
4.5.4 Comparison with Other Protocols 79
4.5.5 Storage Overhead 81
4.6 Related Work 82
4.7 Conclusions 83

5 Security Context Router **85**

5.1 Introduction 85
5.2 System Description 88
5.2.1 Security Context 88
5.2.2 The Security Context Router 88

5.2.3 Multi-tier and Overlay Networks 89
5.3 System Model 91
5.3.1 Network Model 91
5.3.2 Mobility Model 92
5.3.3 Edit distance 97
5.3.4 Pattern Classification 97
5.4 Design of SCR 99
5.4.1 Data Sets with Access Functions 99
5.4.2 Functional Modules 102
5.4.3 Reduction in Storage Overhead 108
5.5 Evaluation 109
5.5.1 Evaluation Methodology 110
5.5.2 Evaluation Results 111
5.5.3 Numerical Analysis 114
5.6 Related Work 117
5.7 Conclusions 120

6 Conclusions and Recommendations 123

6.1 Conclusions 123
6.2 Recommendations for Future Research 124
6.3 Closing Remarks 125

7 Acronyms 127

List of Figures

1.1 An authentication scheme for the integrated networks 8

2.1 An example of relevant protocol stacks in the IEEE 802.1X 19
2.2 EAP message exchange 20
2.3 An example of the 4-way handshake 22
2.4 An example of preauthentication 24
2.5 Proactive key distribution 25
2.6 An integrated authentication system 28
2.7 An example of successful SIM-based mobile authentication 29

3.1 Inter-domain authentication via Diameter over UMTS 36
3.2 Comparison of authentication latency 37
3.3 Increase in authentication latency as the distance grows 38
3.4 Mobile authentication scenario 39
3.5 An abstract model for quantifying the cost of message exchanges 40
3.6 Comparison of cost with various gs as l grows when $N = 10$ 44
3.7 Evaluation of optimal position for the deputy server ($N = 10$) 45

4.1 Message exchanges in the IEEE 802.11 and .11i systems 50

4.2 Message exchanges for a remote access grant in Kerberos 51
4.3 An example of message exchanges in symmetry-key-based NS protocol . . 53
4.4 Authentication architecture 58
4.5 Defined keys hierarchy and boundary 60
4.6 The simulation model for inter-domain handoffs 75
4.7 Authentication latency variations in different configurations of foreign servers 77
4.8 Cumulative distributions of authentication latency 78
4.9 System storage availability affected by the authentication occurrence ratio . 79
4.10 End-to-end domain distance vs. authentication latency 80
4.11 CPU utilization 81
4.12 Latency comparison of MAP with MNS and Kerberos 82

5.1 Multi-tier access and overlay networks 91
5.2 A grid of access networks 92
5.3 Anchor-based determination of directions 93
5.4 Comparison of a variety of distributions 96
5.5 An SCR architecture 100
5.6 Estimate of forwarding 105
5.7 The effectiveness of ahead-of-time forwarding 112
5.8 Comparison of pattern recognition with various distributions 113
5.9 Pattern classification and predictions 115
5.10 Soft state vs storage overhead 116
5.11 The results of statistical estimate from four plausible distributions 118
5.12 The variance of storage gain 119

List of Tables

4.1 Throughput of hash/symmetric and asymmetric algorithms 74

5.1 Comparison of a weight-imposed method with Liu's and generic methods . 106

5.2 Parameters used in evaluation 110

Chapter 1

Introduction

1.1 Motivation

Inter-wireless technologies, ranging from IEEE 802 networks, such as Wi-Fi, WiMax and personal area network to non-802 networks, such as cellular networks, are rapidly converging. This allows mobile users carrying a multimedia-access device to roam across inter-technology-based networks. For instance, a mobile user currently associated with a cellular network can move and switch into wireless local area networks (WLANs) and vice versa. Time-sensitive applications, such as Voice over IP (VoIP) or video streams, are now possible over WLANs such as those based on the IEEE 802.11 Standard [7]. We anticipate, furthermore, that mobile users will soon be able to cross the border of different domains without disrupting their on-going application sessions. We use the term 'domain' to refer to a functionally independent group of components which form an administration unit or provide a particular wireless technology access service.

Security concerns are of paramount importance to widespread seamless mobility. The handoff performance correlates with inter-wireless link switch latency and network layer

latency — with security being required at both levels. Significant research [56, 57, 66, 75] on reducing link switch latency has been done. In contrast, authentication latency varies with mobility. When the mobile user moves around near the centralized server the latency can be minimal, while roaming away from the server can result in considerable latency. The authentication latency has a significant impact especially on the link switch phase in the case of cross-domain mobility because it requires remote contact with a home authentication server. An application for VoIP, for instance, requires the completion of a handoff in less than 50*ms* for acceptable Quality-of-Service (QoS) [76]. Note that the execution of a securing phase is essential as part of a secure handoff mechanism. When the technology curve regarding link switch latency reaches the limit, the authentication latency will become a dominating factor. Clearly, providing a solution to minimize the latency impact without compromising high-level security is a big challenge.

When the mobile node (MN) crosses the domain boundary, authentication latency problems involve a number of factors ranging from the authentication scheme to linking mechanisms between heterogeneous end-systems. For example, message signaling latency requires inter-domain authentication schemes, the efficiency of relevant authentication protocols, and security-relevant resource allocation among heterogeneous end-systems. An MN registers initially with its *home* authentication server. The server then creates the MN's *credential* including private information for *authentication*, *authorization* and *accounting* (AAA) [1]. When the MN roams to a foreign domain, it is bound to the foreign server. An authentication request is forwarded to the MN's home server which then verifies its identity (ID), whereas normal authentication requests in the home domain are handled locally without contacting any remote server. The remote contact, caused by the MN's cross-domain handoff, incurs a significant authentication latency that is part of the handoff latency. The need to avoid such remote contact has been stated in [49].

To sum up, the target environment we consider is the wireless access networks combined with WLANs and cellular networks. In such an environment, when the MN crosses the domain boundary and roams around in a foreign domain with ongoing sessions running on the mobile device, the efficiency of the cross-domain authentication system is critical for the real-time performance of the system. In this thesis we present a high-performance authentication architecture to address such latency problems, supporting secure, cost-effective, and scalable cross-domain handoffs.

1.2 Overview of Mobile Wireless Security

Two technologies (i.e. WLAN and cellular networks) specify an authentication mechanism. The WLAN authentication system is based on the IEEE 802.11i Standard [12], while the global system for mobile communications (GSM) uses a subscriber identity module (SIM) for authentication (vs. the universal mobile telecommunications system (UMTS) which uses a universal SIM (USIM)). The following is a brief overview of each mechanism, the details of which will be presented in Chapter 2.

The IEEE 802.11i is an improvement on the IEEE 802.11 Standard specifying security mechanisms for WLANs. The goal of the 802.11i authentication is to secure a wireless link that will be established between a legitimate mobile device and access points. The *supplicant* in the mobile device sends the request via an extensible authentication protocol (EAP) to the *authenticator*, residing on the access point. The authenticator encapsulates and forwards the request via a specified back-end authentication protocol to the authentication server. In the server, the supplicant's ID is verified in cooperation with an EAP-based authentication protocol. The supplicant and server exchange a series of messages via the authenticator. In case of a successful authentication, they generate a pair-wise master key

and, in particular, the server transfers the key to the authenticator via a secure channel e.g. IPsec [42, 78]. In turn, the authenticator and supplicant generate a pair-wise temporary key, based on the pair-wise master key with which to protect their communication link.

There are three important units in the authentication of cellular networks [60]: a home location register (HLR) is a database used to store permanent data about subscribers, their service profile, location information and activity status; a visitor location register (VLR) is a database that contains temporary information on subscribers and when the mobile user roams into a new area it contacts an appropriate HLR; and an authentication center (AuC) provides authentication and encryption parameters. Authentication uses a challenge-response mechanism. A SIM (USIM) runs on the mobile user's device for the authentication of the GSM (UMTS). The SIM stores network state information such as the mobile user's current location area ID. Each SIM is uniquely identified by its SIM serial number, i.e. a 19 or 20 digit unique number identifying an individual SIM card. Note that SIM-based mechanisms authenticate mobile devices rather than mobile users.

1.3 Requirements

The following is required to design a high-performance authentication architecture for cross-domain mobility.

- Signaling latency in authentication requests must be minimal. The conventional authentication architecture, i.e. a client-server model, requires all requests to eventually be processed by a single server. In such a scheme, unexpected authentication failures, even for legitimate users, may occur due to authentication traffic congestion. Moreover, remote requests generated in a foreign domain incur a fundamental delay that may increase in proportion to geographical distance between

the client's home domain and foreign domain. Therefore, the message delays, due to geographical distance transmission delays, must have a minimal impact on fast handoffs with an acceptable QoS. A means to minimize these delays is to use an appropriate local 'agent' as a 'proxy' for the remote authentication server. However, such agents, as opposed to current AAA-defined agents, are expected to undertake part of the server's authentication policy. In this context, how efficiently the agents are organized is important for high-performance handoffs.

- An authentication system operates in coordination with an authentication protocol defined by the particular service. When the mobile users cross domain boundaries, an appropriate authentication protocol that supports *inter-domain* authentication is essential. This implies the need for interaction among the mobile user, the server in the visited domain, the server in the domain in which the mobile user has previously resided and the home server. Moreover, such protocols must also support mutual authentication because the mobile user needs to ensure that the corresponding server in a foreign domain is legitimate. Conventional authentication protocols cannot be applied straightforwardly to inter-domain authentication. In such an inter-domain authentication, the protocol requires an efficient message-handshake mechanism since cross-domain message signaling is critical to the performance.

- The authentication server authenticates the mobile, based on its credentials. Provided that security information derives from the credentials and is used to authenticate it the next time, one can avoid referring to the credentials by propagating the replicas of security information ahead of time to the server's *neighbors* in adjacent domains. However, propagating the replicas greatly increases storage overhead in the neighbors. A solution to address the storage overhead is to

delete obsolete ones at each regular unit of time. The more frequently it verifies, the lower the storage overhead, but it may weaken the effectiveness of propagation; an optimal value of threshold is required to be determined. In addition to the storage overhead, consistency in the propagated replicas must be maintained.

1.4 Defined Entities

The following items are defined and used throughout the thesis.

- Mobile (user): carries a mobile device, referred to as mobile station (STA) in cellular networks. The term mobile node (MN) is used interchangeably in IP-based networks. The mobile user or node is an authenticated object as well as authenticating subject in WLANs, while the STA is an authenticated object in cellular networks.

- Authentication server: provides mobile users or devices with authentication services. It also has to forward requests from roaming users/devices to other authentication servers. In addition to the authentication services, the authentication server often generates a pair-wise key to support confidentiality. The key generation may be provided in coordination with other security protocols like TLS or PKI.

- Authenticator: the WLAN access point that relays authentication traffic to and from an authentication server. It is an end-point that eventually establishes a secure link with the supplicant.

- Security context: MN's *credentials* including private information for AAA [1]. It includes information on mobile user ID, various encryption keys, validity time and so forth. The need to avoid remote contact to access and verify this information has

been stated in [49] and the concept of security context has subsequently been introduced in [41]. A visiting MN's security context is used by the foreign server to authenticate the MN locally.

- Deputy agent: involved in the authentication process in coordination with the authentication server. It makes use of security context to authenticate the MN. The agent's part in the process is to send a capability signal as detailed in Chapter 3.

- Security context router: a self-organized entity that extends the deputy agent. It manages security contexts. Multiple security context routers are interconnected to form an independent network as detailed in Chapter 5.

1.5 What We Achieved

We designed a high-performance authentication architecture by extending the AAA scheme, basically supporting the integration of WLANs and cellular networks. The integrated network can also include other wireless technologies such as WiMAX and Bluetooth personal area network. Figure 1.1 illustrates a prototype of the architecture that represents the integrated network. The multiple agents that may be (extended) deputies are connected to each other in a peer-to-peer manner, while each controls a sub-tree in a hierarchical manner — the agents may reside on the same physical machine as the server. In this architecture, the authentication requests are processed in distributed agents. Given an authentication protocol that supports inter-domain authentication, collaborating agents enable authentication efficiently by sharing security information. The security information is transferred among the agents; the peer-to-peer-based scheme does not reflect geographical topologies of end-systems. In consequence, an optimized structure, authentication protocol and secu-

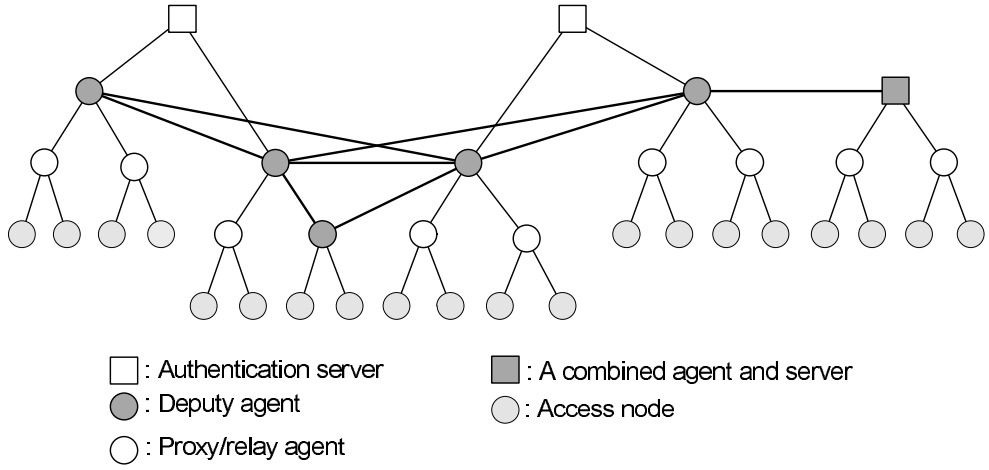


Figure 1.1: An authentication scheme for the integrated networks. The authentication servers are responsible for a cellular network and/or WLAN. The deputy agents are connected to each other. The mobile users are associated with the access nodes. There possibly exist proxy and relay agents between the access nodes and servers.

rity information management characterize the architecture we designed.

1.6 Main Contributions

The following are the three main contributions made by this thesis:

- *Decentralized Authentication Scheme:* Conventional authentication schemes, based on a client-server model, are improved for cross-domain mobility. The current AAA protocols define several agents, including proxy and relay agents, but they do not enforce authentication and authorization. Authentication incurs a round trip time (RTT) due to transmission delays that are proportional to the distance between the server and clients. We define a deputy agent (referred to as an AAA broker in [43]) that performs an important role in the authentication process on behalf of the server; it differs from the server mostly in that the server has knowledge of the clients' permanent information containing its secret key. Instead, the deputy agent is

responsible for authenticating the MN using the security context pre-fetched from the server. The hierarchical authentication scheme with the deputy agents mitigates the authentication latency dramatically [23, 45].

- *Mobility-adjusted Authentication Protocol (MAP):* We propose a cross-domain authentication protocol [47]. It functions in two discrete ways. Initially, the mobile client and the server are mutually authenticated from scratch. This procedure results in producing a security context, including ID, pairs of temporary keys, authentication codes and random numbers, validity, and other information. Next, MAP accomplishes minimum handshake in coordination with deputy agents, which contributes to a further reduction in the authentication latency. In contrast to Kerberos [48] which favors inter-realm authentication, MAP achieves a significant reduction in the authentication latency without degrading high-level security.

- *Security Context Router (SCR):* This efficiently manages the security context to avoid remote contact with the home server, and is designed as an extension of the deputy agents. The key feature of the SCR is to provide predictive forwarding of the security context, characterized by approximate pattern matching and statistical estimation methods. The main contribution is a combination of effective algorithmic techniques for the acceleration of the handoff procedure, efficient estimation and security context transfer between the pairs of collaborating SCRs, which keep storage overhead to a minimum. Via experiments on a prototype implementation and numerical analysis, we demonstrate the performance benefits of the SCR using two methods and determine how well they lower storage overhead.

1.7 Thesis Outline

The remainder of this thesis is organized as follows.

Chapter 2 surveys the state of the art for WLAN and cellular network technologies and their authentication mechanisms, i.e. the IEEE 802.11i and (U)SIM. First, the IEEE 802.11i comprises the IEEE 802.1X that operates in coordination with the AAA and EAP-based protocols, the 4-way handshake and group-key handshake. Second, in addition to the authentication mechanisms, this chapter introduces the related work on fast authentication, including preauthentication and proactive caching of keys. Last, it describes authentication mechanisms (i.e. SIM and USIM) in the cellular network, and approaches to the integration of WLAN and cellular networks.

Chapter 3 concerns a hierarchical authentication scheme by introducing a deputy agent. First, it presents the effectiveness of the deputy agent via an experiment conducted with the implementation of an AAA-based UMTS application. The scheme is then generalized into a decentralized one that is represented as a mathematical abstract model. Lastly, this chapter quantifies the authentication latency based on the model, and evaluates the performance of the scheme.

Chapter 4 proposes a mobility-adjusted authentication protocol (MAP) for cross-domain mobility. MAP is characterized mainly by mutual authentication and pair-wise key generation. It enhances the performance of the IEEE 802.11i authentication. First, the chapter gives an overview of relevant systems i.e. the IEEE 802.11i mechanism, cross-domain-related protocols, and prerequisites of BAN logic. We then design MAP, including architecture, defined keys and messages exchanged, and give a detailed description of elementary modules. Afterwards, security concerns are discussed by first proving MAP in BAN logic and observing a variety of possible attacks. The performance of MAP is evaluated,

compared with the two protocols presented originally.

Chapter 5 concerns the design of a security context router (SCR). A group of SCRs form an overlay network and manage the security context. This chapter first describes the overall system on which the SCR is eventually built and then defines a mobility model that represents the movement direction of the MNs. We then design the SCR and elaborate on its two key features in predictive forwarding. In addition to these features, the functions of garbage collection and connectivity are described, and an analysis of lowering storage overhead is undertaken. The performance of the SCR is evaluated via simulation and numerical analysis.

Finally, Chapter 6 concludes this thesis with a discussion of possible future research directions, together with closing remarks.

Chapter 2

The State of the Art

In this chapter we present the security mechanisms defined and used in the main wireless technologies, i.e. WLAN and 3GPP, showing their algorithms and main characteristics. We also discuss their limitations in such circumstances as when the mobile node (MNs) intend to move across domains with no disruption of their ongoing sessions.

2.1 The IEEE 802.11i Standard

Security mechanisms for IEEE 802.11 networks were standardized in 2004, including a definition of wired equivalent privacy (WEP) for backward compatibility with the original IEEE 802.11 Standard, 1999. The IEEE 802.11i [12] is an amendment to the IEEE 802.11 standard specifying security mechanisms for wireless networks because WEP was shown to have severe security weaknesses. Its main goal is to provide robust security associations for 802.11 networks. The IEEE 802.11i introduces the concept of a security association into wireless networks and defines security association management protocols. More specifically, the IEEE 802.11i architecture is composed of the IEEE 802.1X for au-

thentication; a robust secure network (RSN), containing the 4-way handshake, for security associations; and advanced encryption standard (AES)-based counter mode with a cipher block chaining-message authentication code protocol (CCMP) to provide confidentiality, integrity and authentication.

The IEEE 802.11i has three main components: mobile nodes (MNs), a set of access points along with authentication clients, and a back-end authentication server. When an MN finds an access point with the highest quality radio channels, a sequence of messages is exchanged: open authentication, IEEE 802.1X authentication, 4-way handshake and optional group-key handshake.

2.1.1 The IEEE 802.11 Association

Open system authentication exists for backward compatibility with WEP that is part of the IEEE 802.11 Standard ratified in 1999. It has a sequence of 2-message exchanges. The first message asserts identity and requests authentication, and the second message returns the authentication result. WEP uses the stream cipher Rivest Cipher 4 (RC4) for confidentiality. The standard 64-bit WEP uses a 40-bit key to which a 24-bit initialization vector (IV) is concatenated to make the RC4 key. A 128-bit WEP key is constructed by combining a string of 26 Hexadecimal (Hex) characters, each of which represents 4 bits, with the 24-bit IV; i.e. $26 \times 4 + 24 = 128$-bit WEP key. However, there are weaknesses in WEP, including the possibility of IV collisions and alteration in packets. The same traffic key must never be used twice because RC4 is a stream cipher. Even if the purpose of an IV, which is transmitted as plaintext, is to prevent any repetition, a 24-bit IV is not long enough to avoid any collision. Borisov *et al.* [22] discovered several serious security flaws in WEP that result in practical attacks, showing that WEP fails to achieve its security goals. Cam-Winget *et*

al. [28] surveyed a variety of shortcomings in WEP, and Fluhrer *et al.* [9] presented several weaknesses in the key scheduling algorithm of RC4 and their cryptanalytic significance. After open system authentication, security parameters that specify encryption algorithms, security policy and other information are exchanged between the MN and access point via association exchanges.

2.1.2 The IEEE 802.1X

In the IEEE 802.1X system the MN and access points primarily correlate with each other; in particular, we refer to applications running on the MN and access points as supplicant and authenticator, respectively. The system is characterized by a port-based network access control. In an uncontrolled port (the authenticator is 'switched off') all traffic is unfiltered, while in a controlled port all traffic is blocked except for protocol data units (PDUs) related to extensible authentication protocol (EAP) [21]. When switched on, the authenticator either authenticates a device or user. In the case of mobile device authentication, it is identified by its media access control address (MAC address) that is pre-registered in the authenticator-associated database. In the case of mobile user authentication, the authenticator forwards EAP-based PDUs to a back-end authentication server for authentication. In the case of successful authentication, the port is opened, so that all PDUs are unblocked; otherwise, they remain blocked.

All three roles, i.e. a supplicant, authenticator and back-end authentication server, are necessary to complete an authentication exchange. A given system can adopt one or more of these roles; e.g. an authenticator and a back-end authentication server can be collocated within the same system, allowing that system to perform the authentication functions without the need for communication with an external server. However, the most common

implementation of this mechanism involves the use of an authentication server that is external to the authenticators.

2.1.3 Authentication Authorization Accounting

The AAA Working Group in IETF [1] has developed requirements for Authentication, Authorization and Accounting as applied to network access, the definitions of which are as follows:

- Authentication refers to the confirmation that a user who is requesting services is a valid user of the network services requested. Authentication is accomplished via the presentation of an identity and credentials. Examples of types of credentials are passwords, one-time tokens, digital certificates and phone numbers (calling/called).

- Authorization refers to the granting of specific types of service (including "no service") to a user, based on their authentication, what services they are requesting and the current system state. Authorization may be based on restrictions, e.g. time-of-day, physical location, or restrictions with respect to multiple logins by the same user. Authorization determines the nature of the service which is granted to a user. Examples of types of service include, but are not limited to: IP address filtering, address assignment, route assignment, QoS/differential services, bandwidth control/traffic management, compulsory tunneling to a specific endpoint and encryption.

- Accounting refers to the tracking of the consumption of network resources by users. This information may be used for management, planning, billing or other purposes. Real-time accounting refers to accounting information that is delivered concurrently

with the consumption of the resources. Batch accounting refers to accounting information saved until it is delivered at a later time. Typical information gathered in accounting is the identity of the user, the nature of the service delivered, when the service began, and when it ended.

Since the AAA architecture is built based on a client-server model, the clients reside on the front-end authentication entity, such as access points or base stations, and send the authentication request. All authentication requests are processed in the AAA home server. The server is, in general, responsible for the authentication process from (part of) a domain and cooperates with the others to support cross-domain mobility. In addition to the client and server, relay, proxy, redirect and translation agents are introduced. Relay and proxy agents forward requests and responses based on routing-related attributes in the protocol and realm routing table entries. Relay agents do not make policy decisions and examine/alter any attributes in the packets to be forwarded, while proxy agents do. However, proxy agents do not respond to client requests prior to receiving a response from the server. In other words, it is possible for only the server to respond to the requests. Redirect agents refer clients to the server and allow them to communicate directly rather than forwarding requests and responses between clients and servers. Translation agents perform protocol translation among other AAA protocols. Consequently, when MNs are within a visited domain, the server in control of the visited domain forwards the requests to the MN's home server. It is clear that the long-distance traffic of message exchanges occurs.

2.1.4 EAP-based Authentication Protocols

EAP [21] is a carrier protocol that carries any security protocols; e.g. transport level security (TLS) [31] provides for mutual authentication and key exchange between two endpoints and is carried over EAP-TLS [13]. In addition to EAP-TLS, there are a variety of EAP-based protocols, like EAP-SIM [36] for the SIM mechanism, EAP-AKA [18] for the zUSIM mechanism, protected extensible authentication protocol (PEAP) [67], EAP-SKE [72], EAP-TTLS [34], and so forth. These protocols must provide an authentication mechanism — most of which also support key generation after successful authentication. They operate, based on a client/server scheme. That is, an EAP peer runs on a supplicant and interacts with a back-end authentication server in the AAA scheme.

Figure 2.1 illustrates the relationship between the supplicant, authenticator and authentication server, as well as the structure of the protocols used. The authenticator's controlled port is basically in the unauthorized state and therefore all PDUs are blocked. However, EAP payloads can *pass through* the authenticator; the communication between the authenticator and authentication server relies on either RADIUS [70] or Diameter [26]. The authentication server either authenticates the supplicant or there is mutual authentication between the authentication server and the supplicant, in both cases using one of the above EAP-based security protocols.

There are 4 types of packets in EAP: Request, Response, Success, and Failure. The authenticator sends the Request packet to the supplicant. Additional Request packets are always sent after a valid Response packet is received. The supplicant must send a Response packet in reply to a valid Request packet, but is not allowed to retransmit it during a given period of time. The Success or Failure packet is sent to the supplicant according to the authentication result. Figure 2.2 shows an EAP-based message exchange flow. After the

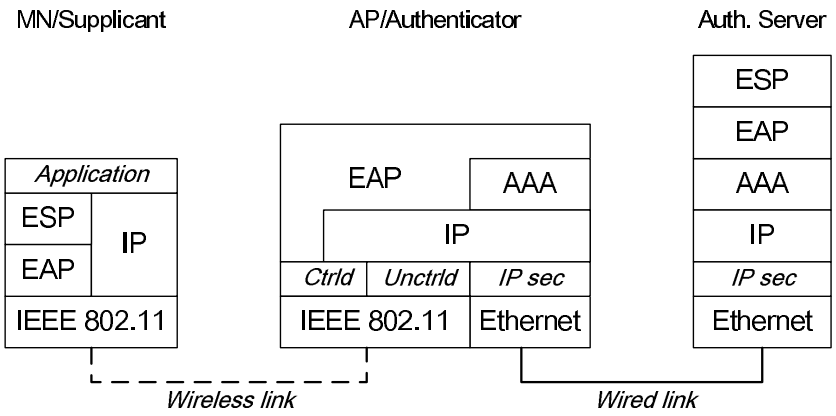


Figure 2.1: An example of relevant protocol stacks in the IEEE 802.1X. E-SP stands for EAP-based security protocols.

initial message initiated by the authenticator, the authentication server undertakes the rest of the exchange. The number of EAP message exchanges is subject to the EAP-based protocols used for either a mutual or unilateral authentication. Alternatively, a supplicant-initiated authentication is allowed; the supplicant sends the authenticator a *trigger* message, of `EAPOL-Start`. The subsequent exchanges are the same as the authenticator-initiated authentication.

2.1.5 4-Way Handshake

A successful authentication in the IEEE 802.1X leads to the generation of a pair-wise master key (PMK) shared between the supplicant and authentication server. The key is eventually used for the 4-way handshake taking place between the supplicant and authenticator. The authenticator receives the PMK from the authentication server via a secure channel.

The IEEE 802.11 uses `EAPOL-Key` frames to exchange information between the supplicants and authenticators. These exchanges yield cryptographic keys and a synchronization of security association state. The following is a description of each message exchanged in the 4-way handshake, of `EAPOL-Key` frames.

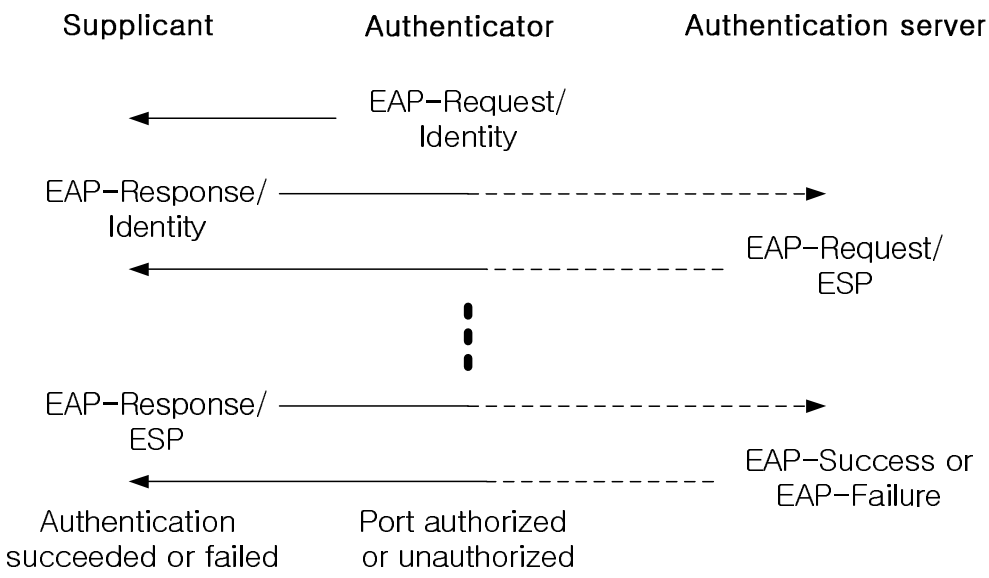


Figure 2.2: EAP message exchange

M1. is sent to the supplicant by the authenticator. It contains the key data field of an encapsulated the PMK identity (PMKID) and a random value that the authenticator generates. If this message is resent, the ID must be unchanged.

M2. is sent to the authenticator as a reply to M1. It contains the key data field of a robust security network (RSN) information element, a random value, and the message integrity code (MIC) of M2. The RSN information element contains authentication and pair-wise cipher suite selectors, a single group cipher suite selector, an RSN capabilities field, the PMKID count and PMKID list. All supplicants implementing RSN association support this element. The size of the RSN information element is limited by the size of an information element which is 255 octets. On receipt of M2, the authenticator verifies that the pair-wise cipher suite selected is one of its configured cipher suites and that the group cipher suite is consistent.

M3. is sent to the supplicant. It contains the key data field of an RSN information

element, a random value and an MIC. If a group cipher has been negotiated, it also includes an encapsulated group temporary key (GTK). The authenticator inserts the RSN information element it previously sent in the Beacon/Probe response message. The supplicant verifies the selected security RSN information with the RSN information element in M3. If the values do not match, the supplicant breaks the association by invoking a `Deauthenticate` request. A security error is logged at this time.

M4. is sent to the authenticator. It contains the key data field that can be empty or can contain an MIC.

During the 4-way handshake, both the supplicant and authenticator generate a PTK. Figure 2.3 shows an example of the 4-way handshake message exchanges.

- The authenticator sends an `EAPOL-Key` frame containing an ARAND. The supplicant derives a PTK from the ARAND and SRAND. That is, it computes PRF-X(PMK, "pairwise key expansion" | Min(AA,SPA) | Max(AA,SPA) | Min(ARAND, SRAND) | Max(ARAND, SRAND)), where PRF-X is a pseudo random function generating X-bit output data, and AA and SPA denote the MAC addresses of the authenticator and supplicant, respectively. Note that the PMK is known only by the supplicant and authenticator.

- The supplicant sends an `EAPOL-Key` frame containing the SRAND, the RSN information element from the (re)association request frame, and an MIC. The authenticator derives a PTK from the ARAND and SRAND in the same manner as the supplicant, and verifies the MIC in the `EAPOL-Key` frame.

- The authenticator sends an `EAPOL-Key` frame containing the ARAND, the RSN

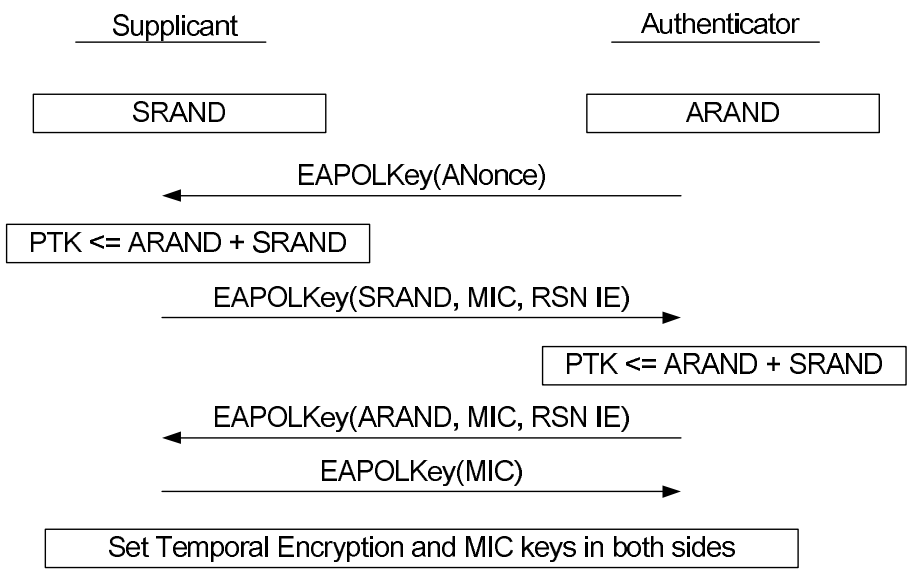


Figure 2.3: An example of the 4-way handshake

information element from its Beacon/Probe response messages, MIC and the encapsulated GTK (if available).

- The supplicant sends an `EAPOL-Key` frame to confirm that the temporary keys are installed.

The authentication server also knows the PMK. Therefore, additional assumptions are required: the authentication server does not expose the PMK to other parties, it does not masquerade as the supplicant to the authenticator or as the authenticator to the supplicant, and it does not masquerade as the supplicant itself or the authenticator. If any of these assumptions are broken, then the protocol fails to provide any security guarantees.

The 4-way handshake uses random values against replay. The ARAND provides replay protection to the authenticator, and the SRAND to the supplicant. In most session initiation protocols, replay protection is explicitly accomplished by selecting a random value and requiring the peer to reflect the received random value in a response message. The 4-way

handshake instead mixes the ARAND and SRAND into the PTK, and replays are detected implicitly by MIC failures.

2.1.6 Group-key Handshake

The authenticator uses the group-key handshake, which is optional, to send a new GTK to the supplicants. The following are the message exchanges:

M1. authenticator → supplicant: `EAPOL-Key`(RSC, MIC, GTK[KeyID])
The authenticator generates, encapsulates and sends a new GTK, along with the last sequence number (receive sequence counter (RSC)). The supplicant verifies the MIC, decapsulates the GTK with KeyID obtained during the 4-way handshake, and configures it.

M2. supplicant → authenticator: `EAPOL-Key`(MIC)
The authenticator verifies the MIC and configures the GTK into the IEEE 802.11 MAC if verification is successful.

If the authenticator does not receive a reply to its messages during a given period of time, it deauthenticates the supplicant, and the supplicant fails to be authenticated.

2.1.7 Fast Authentication

Preauthentication

Preauthentication as part of the IEEE 802.11i can be useful for a performance enhancement of handoffs. Figure 2.4 depicts a sequence of a preauthentication process. Before preauthentication, the new AP, with which the MN will be associated, advertises the preauthentication capability in the RSN information element. An MN's supplicant initiates preau-

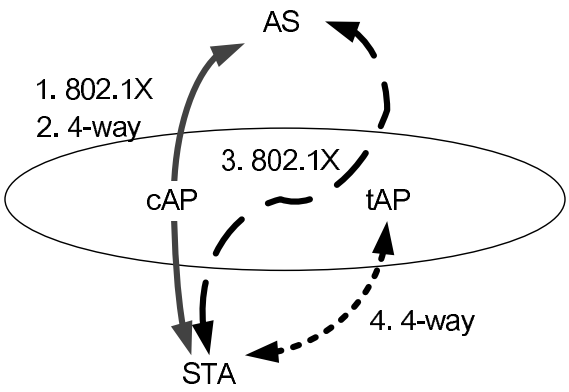


Figure 2.4: An example of preauthentication. Current and target APs (i.e. cAP and tAP) are in the boundary of an authentication system. Steps 1 and 2 are assumed to be completed before preauthentication starts.

thentication after completing the 4-way handshake and configuring the required temporary keys. It sends the authenticator of the target AP the IEEE 802.1X EAPOL-Start message with the addresses of the APs. The IEEE 802.1X authentication takes place and a successful authentication results in deploying a PMK security association (PMKSA) to the MN and preauthenticated AP. Then, when the MN associates with the preauthenticated AP, the supplicant uses the PMKSA to perform the 4-way handshake. Unless the PMKSAs of the supplicant and authenticator are matched, the 4-way handshake fails and the IEEE 802.1X authentication is performed from scratch. The main drawback of preauthentication is that it is highly uncertain that the MN will be associated with the preauthenticated AP.

Proactive Caching of Keys

One can achieve fast authentication with proactive key distribution [58]. Figure 2.5 depicts how the AS propagates PMKSAs to relevant APs with which the MN may be associated. The AS builds neighbor graphs that represent associations among APs [57]. After the MN completes the 4-way handshake, the AS propagates PMKSAs to the selected APs, referring to a neighbor graph — there is a variant, i.e. a selective neighbor caching scheme [66]

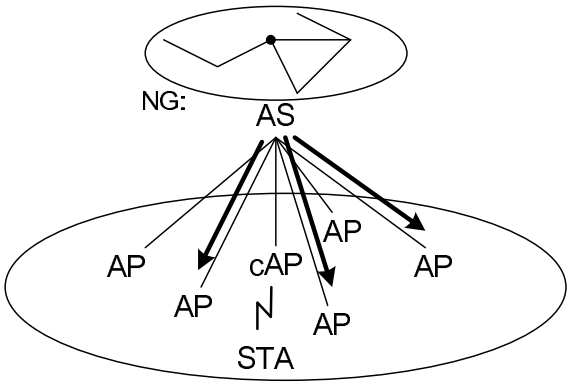


Figure 2.5: Proactive key distribution. cAP is an AP with which the MN is currently associated. The AS propagates the derived PMKs to APs selected based on a neighbor graph (NG).

that propagates a subset of neighbor APs, distinguishing the probability of each AP with which the MN will be associated. Each PMKSA is computed by hashing a combination of MK, the old PMKSA, and the addresses of the MN and each AP, i.e. PMKSA_n=H(MK, PMKSA_{n-1}, MN's MAC Address, AP's MAC Address). When the MN is associated with one of the APs that has PMKSA pre-deployed, the MN and AP perform the 4-way handshake directly without the IEEE 802.1X. If their PMKSA is different, the IEEE 802.1X runs from scratch. The neighbor graphs can also be managed in each AP. If so, then PMKSA is propagated via IAPP [11].

2.2 Authentication in 3GPP

The authentication mechanisms in the 3G partnership project (3GPP) [10] are divided into UMTS authentication and GSM authentication. UMTS authentication is based on a USIM for authenticating the MN and network (e.g. authentication server (AuC)) — it implies mutual authentication in contrast to GSM authentication, while GSM authentication is based on a SIM or GSM-capable USIM for authenticating the MN. GSM authentication

uses a triplet of parameters: a network challenge (RAND), an expected response value (SRES), and cipher key (Kc); UMTS authentication uses a quintet of parameters: RAND, an expected response (XRES), cipher key (CK), integrity key (IK) and authentication token (AUTN).

The GSM network authenticates the international mobile subscriber identity (IMSI) through the use of a challenge-response mechanism. A 128-bit RAND is sent to the MN. The MN computes the 32-bit SRES based on the encryption of the RAND with the authentication algorithm (A3) using the personal identity number (PIN). Upon receiving the SRES, the GSM network repeats the calculation. If the received SRES and the calculated value match, the MN is successfully authenticated. Otherwise, the connection is terminated due to an authentication failure. Similarly, the UMTS network uses a challenge-response mechanism. In contrast, it sends a challenge message additionally containing the AUTN. Upon receiving this message, the USIM in the MN verifies the AUTN, and if it is accepted, the USIM continues to compute the signature of RAND and RES, and then sends RES to the UMTS network while computing a new CK and IK. The UMTS network compares the received RES and XRES. If they match, a mutual authentication is successfully achieved and both the UMTS network and MN eventually generate the Kc from the CK and IK.

2.3 The Integration of WLAN and GPRS

WLAN can complement mobile operators' traditional wide-area General Packet Radio Service (GPRS) by offering a cost-effective wireless broadband data solution indoors. Several approaches have been proposed for interworking between WLANs and cellular networks. The European Telecommunications Standards Institute (ETSI) has specified two generic approaches i.e. *tight coupling* and *loose coupling* [33]. With tight coupling, the WLAN is

connected to the GPRS core network via serving GPRS support node (SGSN) and treated as other radio access networks (RANs), such as GPRS RAN and UMTS terrestrial RAN. Therefore, the WLAN data traffic flows through the GPRS network before reaching IP-based networks. This scheme is primarily tailored to support WLANs operated by cellular operators, and thus hardly supports third-party WLANs. On the other hand, with loose coupling, the WLAN reaches IP networks via an operator's IP network, and is thus connected to the GPRS network via the gateway GPRS support node. The solution to this scheme relies substantially on IETF protocols; i.e. Ala-Laurila *et al.* [17] presented a new WLAN system architecture that combines WLAN technology with mobile operators' SIM-based subscriber management functions and roaming infrastructure. Salkintzis *et al.* [73] compared the two internetworking mechanisms and discussed their advantages and drawbacks. In particular, an authentication system in an integrated network with loose coupling is based on AAA protocols, such as RADIUS and Diameter.

Figure 2.6 depicts an integrated authentication system via the AAA architecture that covers WLANs and cellular networks. If the MN subscribes to a cellular operator, then the AAA associated with the cellular network becomes its home server (AAAH). When the MN intends to have access to a WLAN, the authentication request is forwarded to AAAH; AAAF plays a role of proxy/relay agent in this case.

Figure 2.7 shows a signaling diagram for a successful SIM-based authentication while the MN roams in a WLAN. At first, the MN's supplicant sends an authentication triggering message, `EAPOL-start`, to an AP's authenticator (e.g. RADIUS client). In messages 2 to 4, the authenticator obtains the MN's ID — which corresponds to IMSI in the case of GSM and UMTS — and forwards the ID to its back-end authentication server (e.g. RADIUS server). The server notices that the MN belongs to another server in a foreign domain by verifying a domain indicator that is usually set by the supplicant. The RADIUS client can

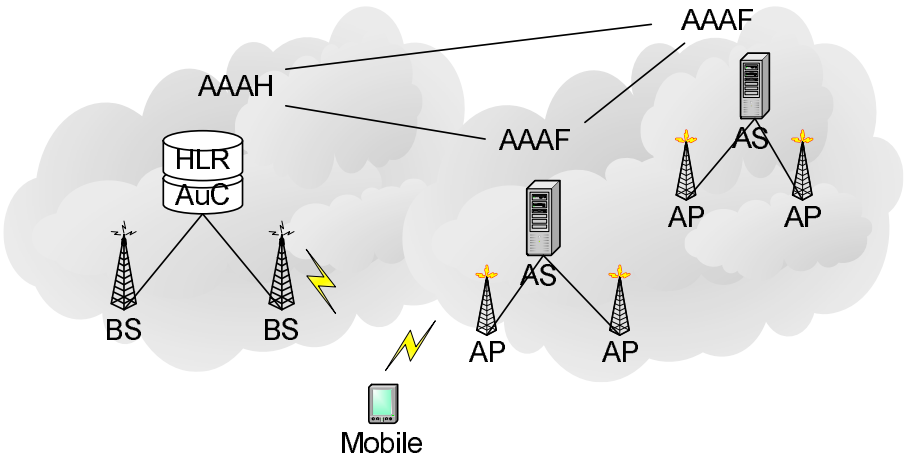


Figure 2.6: An integrated authentication system covering WLANs and cellular networks

also attach the domain indicator to which this request will be forwarded, and to do so, pre-configuration in the client is required. When receiving message 5, the MN's home server (e.g. RADIUS server) fetches authentication information, i.e. GSM authentication vectors (UMTS authentication vectors for UMTS) by sending the MN's IMSI (in messages 6 and 7). In messages 8 and 9, the home server sends a random-challenge message, forwarded by the foreign server, to the authenticator. In messages 10 and 11, the MN's supplicant computes an SRES with the received RAND and sends the authenticator the SRES. In messages 12 and 13, the home server matches the SRES received via the foreign server with the XRES. The authentication result is transferred to the MN eventually in messages 14 to 16 (it is a successful authentication in the figure; otherwise `RADIUS access-reject` and `EAP-failure` are sent). Afterwards, the 4-way handshake or WEP security can be performed depending on what security mechanism is used to protect the WLAN. Each time the MN roams around in a foreign domain, the above steps are repeated. Therefore, in terms of the performance issue, such authentication mechanisms must be improved.

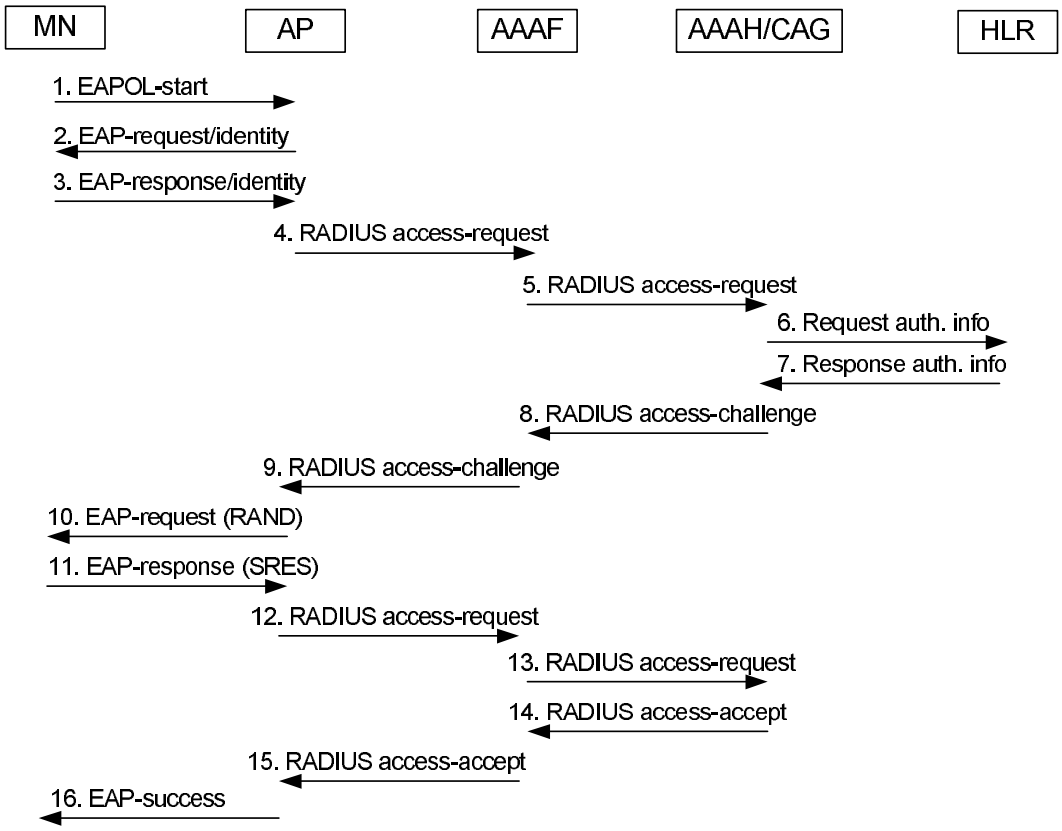


Figure 2.7: An example of successful SIM-based authentication with the MN roaming in a WLAN foreign domain. AAAF and AAAH are RADIUS servers in the WLAN and GPRS, respectively. CAG stands for cellular access gateway that is an interface between IP-based network and GPRS. HLR manages subscribers' credentials.

Chapter 3

Decentralized Authentication Architecture

3.1 Introduction

The server in the AAA scheme verifies the MNs' ID, authorizes them to use resources and enforces a security policy on their profile. The server is passive: it waits for requests and sends a reply. The clients, on the other hand, send requests and wait until replies arrive. Such a scheme fundamentally suffers from authentication failures caused by authentication request congestion regardless of MNs' legitimacy, and incurs significant latency in authentication with the MNs roaming in a visited domain.

Since the proxy and relay agents defined in the AAA framework are not allowed to enforce the authentication policy, we introduce a *deputy agent* that authenticates the MNs on behalf of the server. After successful authentication, the server generates and transfers security information to a deputy agent, so that the deputy agent can process consecutive requests while the MNs are roaming. First, we present an implementation of an AAA-combined

UMTS network authentication via which we demonstrate the impact of the deputy agent on authentication latency. The application of the AAA-combined UMTS network authentication (called AAA-UMTS application) generates a set of authentication elements that are included in security information. In addition to the application, we build an abstract model that represents a decentralized scheme that consists mainly of multiple deputy agents, and quantify the authentication latency in message signaling. The evaluation analysis results in determining which deputy agent best handles the authentication process.

The remainder of this chapter is organized as follows. Section 3.2 presents the implementation of the AAA-UMTS application, including command codes and defined attribute values, and describes a UMTS authentication procedure. It then evaluates the performance results of the experiment conducted, with the implementation in a UMTS platform. Section 3.3 details a decentralized authentication scheme with multiple distributed deputy agents, defines an abstract model of the scheme, and quantifies authentication latency in terms of message exchange cost. Section 3.4 evaluates the performance of the model and Section 3.6 concludes this chapter.

3.2 AAA-UMTS Application

Command Codes

We use the same AA-Request (AAR) and AA-Answer (AAA) messages that are defined in the Diameter network access server application [27]. The messages are captured to apply the UMTS authentication mechanism according to the presence of the attribute value of `UMTS-Proxy-Capability`. A description of the mechanism is given below.

Defined Attribute Values

- `Challenge-Request` is of type Enumerated and contains the request-type identifier determining if it is for a request or response.

- `UMTS-Proxy-Capability` is of type OctetString and contains one octet identifying that there exists a deputy agent capable of serving, as an auxiliary AAA server as described above.

- `UMTS-Vector` is of type Grouped and contains `User-Name`, `REND` and `XRES`. It may appear in the response to challenge. If no deputy agent is able to handle the attribute value (AV) pair, the only `REND` is sent to ordinary proxy agents. Its data field has the following ABNF grammar:

```
UMTS-Vector ::= < AV header >
                { User-Name }
                { REND }
                { XRES }
               *[ AVs ]
```

- `User-Name` is of type UTF8String and contains an ID. `REND` is of type OctetString and contains 16 octets with the 'P' protection bit enabled. `XRES` is of type OctetString and contains 16 octets with the 'P' bit enabled.

- `Token-Response` is of type OctetString and contains 16 octets with the 'P' bit enabled. It is computed and sent to the deputy agent.

UMTS Network Authentication

Provided that the MN visits the UMTS domain, it requests authentication by sending its ID. The AAA client captures this message.

M1. Client → Server: AAR (`User-Name`, `Challenge-Request`, `UMTS-Proxy-Capability`)

On capturing the request, the AAA client sends an AAR message to the home server via several deputy agents containing the `Challenge-Request` AV, indicating that a challenge token is required. Any one of the deputy agents may add the `UMTS-Proxy-Capability` AV to the message, which signifies that it joins the authentication mechanism.

M2. Server → Deputy: AAA (`User-Name`, `UMTS-Vectors`)

The server verifies the presence of the `UMTS-Proxy-Capability` AV in the message, and if several AVs are found, the server chooses an appropriate agent to handle the request via analysis that will be detailed in Section 3.3. In the experimentation version used for evaluation, however, the one in the domain in which the request was originally issued is chosen. The server generates a random value (REND) and computes expected response (XRES) by applying HMAC algorithms [50] with the REND and an MN-shared key. It sends the AAA message, including User's ID and multiple `UMTS-Vectors` of `REND` and `XRES` AVs. In case of no `UMTS-Proxy-Capability` AV found in the AAR message, only a `REND` AV is included in the message.

M3. Deputy → Client: AAA (`User-Name`, `REND`)

The chosen deputy agent extracts the REND from the AAA message and then sends it to the MN to be challenged.

M4. Client → Deputy: AAR (`User-Name`, `Challenge-Request`, `Token-Response`)
The MN computes an RES with the shared key and the received REND, and responds with the AAR message, including the `Token-Response` AV containing the computed RES and the `Challenge-Request` AV set to a value signifying response.

M5. Deputy → Client: AAA (`User-Name`, `Response`)
The deputy agent verifies if the received RES is matched with the corresponding XRES, and then replies with a success/reject message. The next request is treated in the deputy agent without contacting the server.

3.2.1 Evaluation of the AAA-UMTS Application

Figure 3.1 illustrates a scenario of the AAA-UMTS application during inter-domain handoffs. We implemented the application based on the Diameter NASREQ application [27], including proxy, relay and deputy agents. A Diameter client is installed in the base station, and the deputy agent serves as a Diameter server in the Sophia domain. The MN registers with a Diameter server in the Star-vthd domain. An alternate Diameter server in the Enst domain is necessary for evaluation. When associated with the base station, the MN issues the request via an ad hoc protocol, typing user ID and password. The Diameter client sends a Diameter-transformed request to the home server in the Star-vthd domain. For experimentation purposes, it can forward the request to the proxy agent in the Enst domain. The Diameter server in the Star-vthd domain responds with AV pairs to the deputy agent, and then the deputy agent can enforce the UMTS authentication policy. Therefore, the long-distance traffic of message exchanges is substantially reduced.

Figure 3.2 shows the impact of a deputy agent on the UMTS network authentication

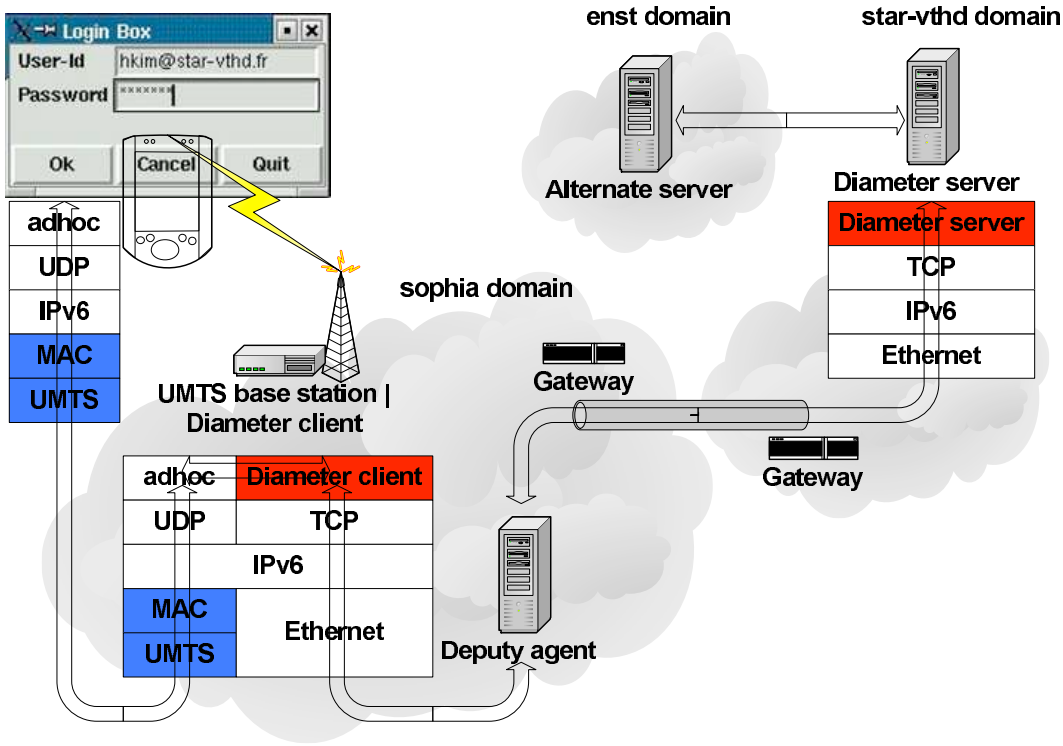


Figure 3.1: Inter-domain authentication via Diameter over UMTS. RTTs between the deputy agent and Diameter server, and between the server and an alternate server correspond to 9.97*ms* and 1.23*ms*, respectively. All associations between the client and deputy, the server and deputy, and the server and the alternate server are assumed to be secure.

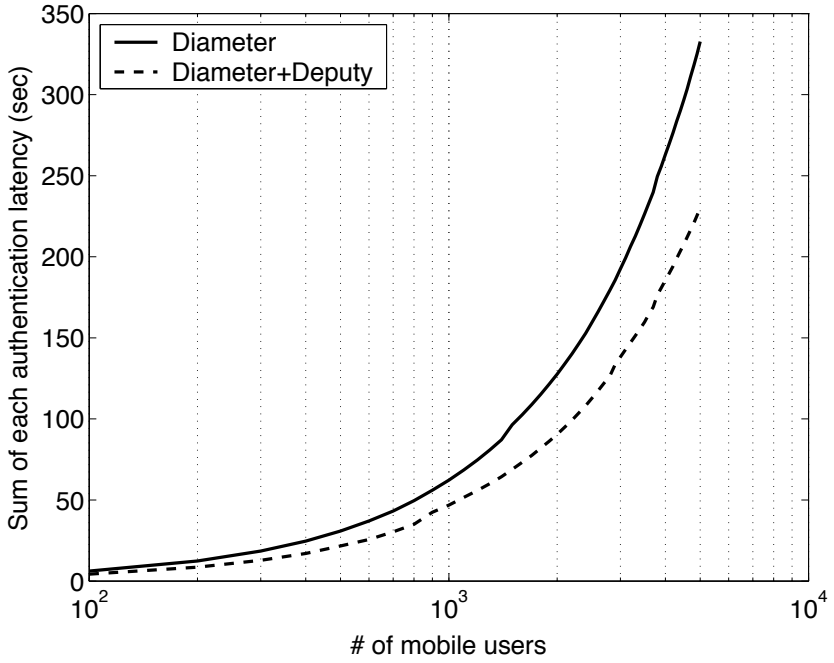


Figure 3.2: Comparison of authentication latency between Diameter and the combination of Diameter and deputy agent as the number of MNs increases. The latency from sending a request to receiving its result is measured.

latency. Authentication in the original Diameter application is processed solely on the server in the Star-vthd domain, while the deputy agent in the Sophia domain authenticates the MN after receiving security information. About 29% of reduction in the authentication latency is achieved in this experiment. The latency gain increases as the MN moves away from the home server.

We install multiple proxy agents in the servers in the Enst and Star-vthd domains, and switch their role from server to proxy, alternately, which emulates the increase in their distance. For example, if 2 proxy agents are necessary, the Diameter server in the Star-vthd domain forwards the request to the server in the Enst domain as a proxy and then processes the returned request as a home server. Figure 3.3 shows the increase in authentication latency as the distance grows. As expected, the latency gain increases proportionally,

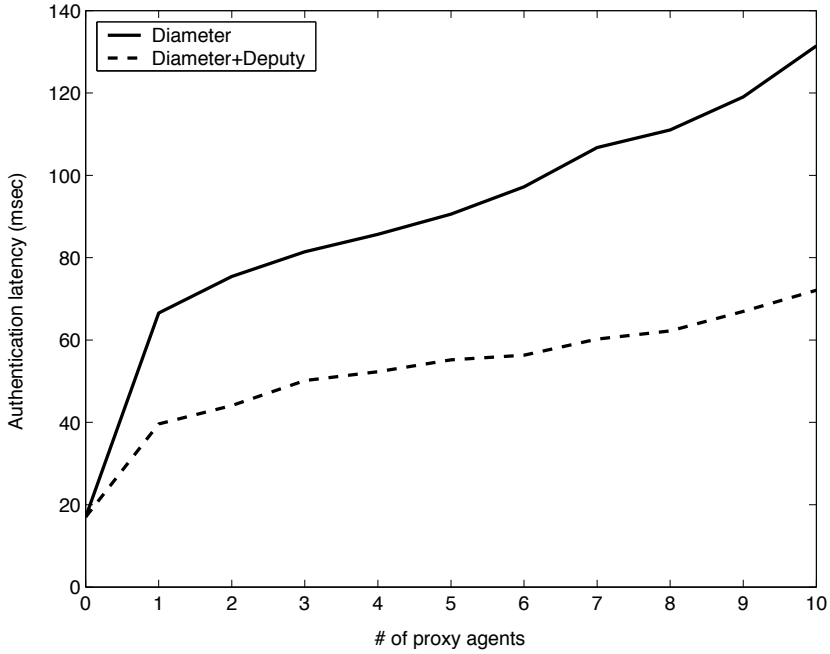


Figure 3.3: Increase in authentication latency as the distance grows

e.g. with 4 proxy agents traversed during the authentication process, 40% latency gain is achieved, and it increases up to 45% with 10 proxy agents.

3.3 Decentralized Authentication Scheme

In an attempt to secure no false negative failures of authentication by the server and the fast cross-domain handoff process, in this section, we introduce an abstract model of the decentralized authentication scheme consisting of multiple deputy agents, and quantify the authentication latency.

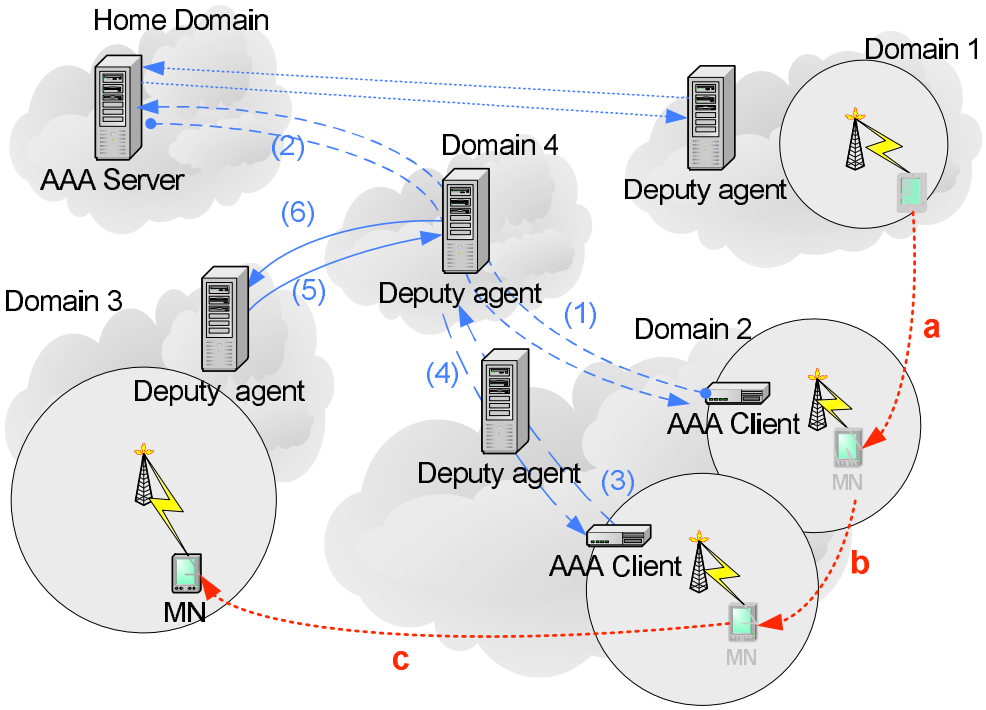


Figure 3.4: Mobile authentication scenario; the connections between deputy agents, between AAA clients and agents/servers, between agents and server are assumed to be secure.

3.3.1 Mobile Authentication

Figure 3.4 illustrates the message flow of authentication during the cross-domain handoff. After crossing the domain boundary, the MN associates with an access point in domain 2, and the request is sent to the home server via two deputy agents adding the joining attribute (e.g. `UMTS-Proxy-Capability` in the AAA-UMTS application). The server determines which deputy agent will undertake the mobile authentication, taking the latency factors into consideration based on an abstract model, which is detailed shortly. Provided that the deputy agent in domain 4 is chosen, the consecutive requests are processed in that agent regardless of the MN handing off into domain 3.

3.3.2 An Abstract Model

Figure 3.5 illustrates a binary-tree-based abstract model. The root and leaves in the binary tree correspond to the AAA server and clients, respectively. An intermediate node is one of deputy, proxy, and relay agents. We assume that all intermediate nodes can play a deputy role in the mobile authentication. We also assume that costs of exchanging a message between all adjacent nodes is the same (i.e. 1 for simplicity); e.g. the cost of sending and receiving back a message to a 2-hop-distance node is equal to 2. The goal in this model is to choose a deputy agent that allows one to keep the cost to a minimum; e.g. if the MN moves around in group A, then n_y may be optimal, and if it moves across groups A and B, then n_x may be optimal.

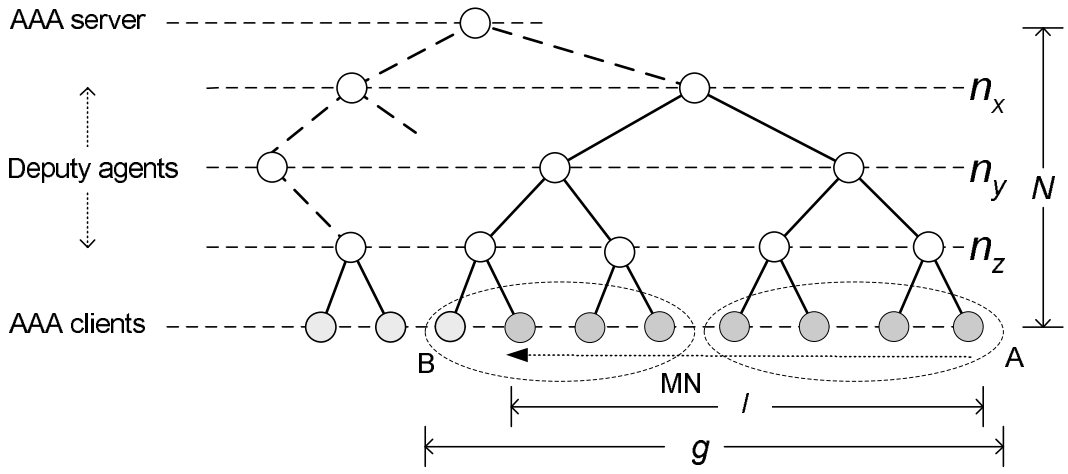


Figure 3.5: An abstract model for quantifying cost of message exchanges. Parameters l and g represent the number of access points that the MN traversed and the number of groups that it visited, respectively. N denotes an end-to-end cost and n_x, n_y, and n_z denote the respective heights of the tree corresponding to the deputy agents' positions.

Cost Function: The following is the derivation of a minimal cost function. When the MN is initially authenticated, the authentication request is sent to the server, which costs N. After a successful authentication, the server calculates a deputy's position that will be

somewhere in the middle. Each time the MN moves with $l-1$ associations (it counts once with a new access point), the request is processed in a chosen deputy agent. Then, the cost with respect to l is

$$(l-1)\frac{\log l}{\log 2}+N. \tag{3.1}$$

If no deputy agent is determined, then the cost is

$$N \times l, \tag{3.2}$$

which corresponds to a non-optimized authentication scheme as well. We divide l into g groups, i.e. $l = l_1 + l_2 + ... + l_g$, assuming that a deputy agent controls each group of clients. The sum of costs with respect to each group is

$$(l_1-1)\frac{\log l_1}{\log 2}+...+(l_g-1)\frac{\log l_g}{\log 2}+gN. \tag{3.3}$$

Here, we define a convex function for an interval $[1, b]$,

$$f(x) = g(x-1)\frac{\log x}{\log 2},. \tag{3.4}$$

Then, Eq. (3.3) is expressed as

$$\frac{1}{g}(f(l_1)+f(l_2)+...+f(l_g))+gN. \tag{3.5}$$

For any two points x_1 and x_2 in that interval, $f(x)$ holds $1/2(f(x_1)+f(x_2)) \geq f((x_1+x_2)/2)$. Therefore, we obtain

$$\begin{aligned} \frac{1}{g}(f(l_1)+f(l_2)+...+f(l_g))+gN &\geq f(\frac{1}{g}\times(l_1+l_2+...+l_g))+gN \\ &= f(\frac{l}{g})+gN, \end{aligned} \tag{3.6}$$

which is a lower bound of the cost function when $l_1 = l_2 = ... = l_g = l/g$. Therefore, the cost function with respect to g is

$$f_C(g) = (l-g)\frac{\log\frac{l}{g}}{\log 2} + gN. \tag{3.7}$$

In particular, when $g = 1$, f_C results in Eq. (3.1), and when $g = l$, it results in Eq. (3.2). Therefore, reducing the cost of authentication requests allows one to adjust g, which in turn produces an optimal deputy's position.

Approximation to Optimal g: We here attempt to compute g that results in minimizing f_C. The derivative of Eq. (3.7) with respect to g is

$$f'_C(g) = N + \frac{-\log(\frac{l}{g})+1-\frac{l}{g}}{\log 2} \tag{3.8}$$

and the second derivative is

$$f''_C(g) = \frac{\frac{1}{g}+\frac{l}{g^2}}{\log 2}, \tag{3.9}$$

which is strictly positive. Thus, f'_C is an increasing function, and there are two cases when $g = 1$ as the basis. (1) If $f'_C(1) \geq 0$, then we deduce that for any $g > 1$, $f'_C(g) > 0$ because $f'_C(g)$ is the increasing function. That is, when $g = 1$, $f_C(g)$ is minimal — it is true only if $f'_C(1) \geq 0$, i.e. $N\log 2 + 1 \geq l + \log l$. (2) If $f'_C(1) < 0$, there exists a value of g satisfying

$f'_C(g) = 0$ which implies

$$N \log 2 = \log(\frac{l}{g}) + \frac{l}{g} - 1. \tag{3.10}$$

Considering that $\log(l/g) \ll l/g$, Eq. (3.10) approximates to $l/g \approx 1 + N \log 2$. Therefore, g approximating to g_{opt} is

$$g_{approx} \approx \begin{cases} \frac{l}{1+N \log 2} & \text{for } N \log 2 + 1 < l + \log l, \\ 1 & \text{for } N \log 2 + 1 \geq l + \log l. \end{cases} \tag{3.11}$$

Latency Gain: The following is a numerical gain in latency from a non-optimized authentication scheme.

$$f_C(l) - f_C(g) = (l - g)(N - \frac{\log \frac{l}{g}}{\log 2}). \tag{3.12}$$

For $N \log 2 + 1 < l + \log l$, the gain is

$$(l - \frac{l}{1 + N \log 2})(N - \frac{\log(1 + N \log 2)}{\log 2}) \approx \frac{l(N \log 2 - \log(N \log 2))}{\log 2} \tag{3.13}$$

and for $N \log 2 + 1 \geq l + \log l$, it is

$$(l - 1)(N - \frac{\log l}{\log 2}) \approx (l - 1)N. \tag{3.14}$$

In both cases the gain increases as $l \to \infty$ or $N \to \infty$.

3.4 Performance Evaluation

A Good Approximation of g: Figure 3.6 shows the performance result from comparing the authentication cost with various values of g. The optimal g generates the best performance

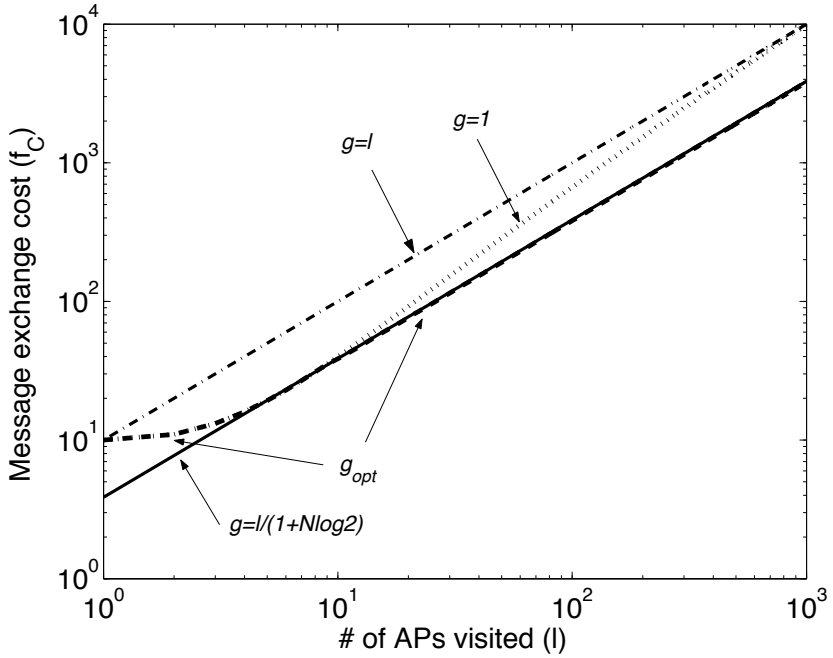


Figure 3.6: Comparison of cost with various gs as l grows when $N = 10$

of reducing the cost as expected. The approximation of g fits g_{opt} perfectly, i.e. when $l \leq 10$, a single deputy agent (i.e. $g = 1$) processes the requests, and when $l > 10$ multiple deputy agents (i.e. $g = 1/(1 + N \log 2)$) do. Provided that the MN associates with access nodes once and is unlikely to return to the visited access nodes, deputy agents located in the AAA clients (i.e. $g = l$) show the worst performance.

Dividing access points into multiple groups performs best. As the MN moves away from its home server, the size of its group increases, which means that a deputy agent located higher than the AAA clients is chosen as optimal. As can be seen in Figure 3.7, the level of the deputy agent's position is raised in the two cases of g_{opt} and g_{approx}. The position is stabilized as the MN traverses the access points further, with 4 or 8 access points grouped and controlled by a deputy agent.

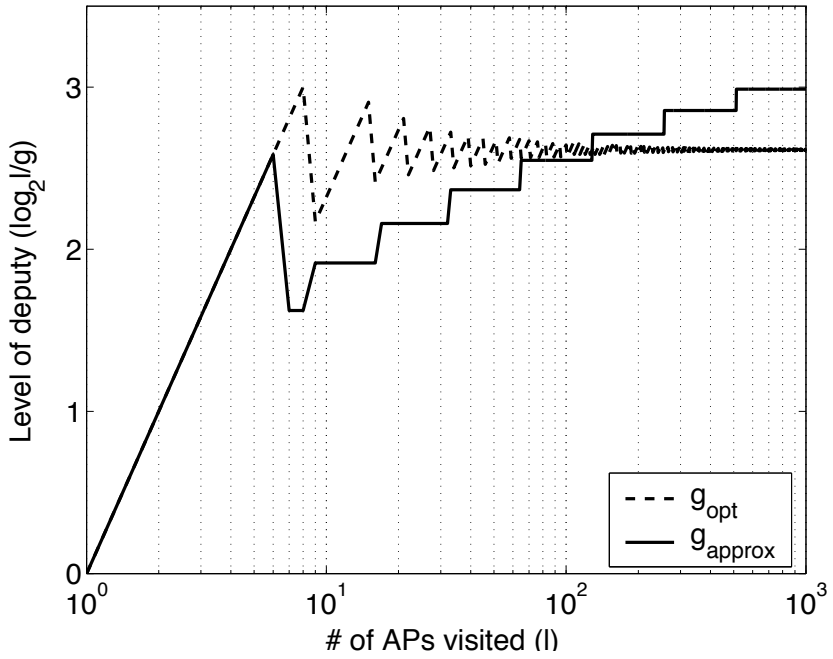


Figure 3.7: Evaluation of optimal position for the deputy server ($N = 10$)

3.5 Related Work

Examples of applications based on decentralized architectures — a peer-to-peer network — include Napster [4], Gnutella [3], and Freenet [2]. In Napster, to retrieve a file, a user queries a central server and obtains the IP address of a user machine storing the requested file. The file is then downloaded directly from the machine, i.e. it uses a client-server structure for searching and a peer-to-peer structure for retrieving files. In contrast, in Gnutella, users self-organize an application-level mesh, i.e. using a peer-to-peer structure for all purposes. CAN [69] allows for a scalable peer-to-peer file distribution system by means of hash table-like abstraction in communication models, which maps keys onto values.

3.6 Conclusions

In this chapter, we presented an implementation of the AAA-UMTS application and confirmed the effectiveness of the deputy agent in authentication latency. We also presented an abstract mathematical model of the decentralized authentication scheme, and quantified the cost of message exchanges to reduce latency. We confirmed that multiple collaborating deputy agents perform better than a single one, by allowing for a reliable and scalable authentication mechanism.

Chapter 4

Mobility-adjusted Authentication Protocol

4.1 Introduction

Time-sensitive applications, such as Voice over IP (VoIP) or video streaming, are now possible over wireless local area networks (WLANs), such as those based on the IEEE 802.11 Standard [7], thanks to their high bandwidth. WLAN technologies also allow mobile users to roam within public/corporate buildings or university campuses. Furthermore, we anticipate that mobile users might cross the domain boundary without their on-going application sessions disrupted. However, VoIP requires a handoff to be completed in less than $50ms$ for acceptable Quality-of-Service (QoS) [76], including the execution of the IEEE 802.11i authentication [12] as part of a secure handoff mechanism.

Minimizing the number of messages to be exchanged is important as cross-domain authentication needs to contact the remote home server. Moreover, authentication latency increases in proportion to the round-trip time between two points involved in inter-domain

message exchanges. Optimization of the authentication protocol is of utmost importance since an existing redundant combination of authentication and key negotiation functions incurs more rounds of message exchange than necessary.

We propose an enhanced protocol for cross-domain authentication, Mobility-adjusted Authentication Protocol (MAP) that relies on far less costly symmetric cryptography. (1) MAP reduces cross-domain authentication latency by reducing the number of message exchanges. MAP requires less message exchanges without degrading security or the re-authentication mechanism, reducing authentication latency significantly. (2) MAP replaces the 4-way handshake of the IEEE 802.11i authentication. In coordination with the authenticator within an access point, MAP defines hierarchical key derivation and generates consecutive keys during authentication operations. This leads to optimizing the 802.11i authentication mechanism by removing the need for the 4-way handshake. (3) MAP leverages the concept of security context to mostly avoid remote contact. With the mobile user moving along, its security context is transferred via security context routers (SCRs) we present in this chapter. An SCR also plays a role of an authentication server in a foreign domain; it provides security context for MAP operating as if in the home server. Via a prototype implementation, our evaluation results show that cross-domain authentication latency of MAP accounts for 74% and 85% that of Kerberos [48] and Needham-Schroeder symmetric-key protocol (NS) [61, 62], respectively. It makes up to 53% improvement in authentication latency which is proportional to the end-to-end domain distance until the round-trip time counts up to 100ms.

The remainder of this chapter is organized as follows. Section 4.2 gives an overview of the 802.11i authentication mechanism, the related cross-domain protocols, design requirements, and prerequisites of BAN logic. Section 4.3 first describes MAP including its architecture and a relevant interaction between SCRs. Subsequently we details defined keys

and types of messages, an example of message exchanges for a successful authentication, and the corresponding pseudo code of each module. Section 4.4 considers possible threats and analyzes the security of MAP. Section 4.5 examines the performance via measurements and simulation. Finally, we discuss related work in Section 4.6 and conclude the chapter in Section 4.7.

4.2 Overview of Authentication Mechanism and Requirements

In this section, we first introduce the 802.11i authentication scheme and protocols applicable to the cross-domain authentication, and then describe the design requirements of authentication protocols. Finally, we explore prerequisites to BAN logic.

4.2.1 The IEEE 802.11i Authentication

The IEEE 802.11i authentication is responsible for mutual authentication and key derivation for securing WLANs via the IEEE 802.1X and 4-way handshake [12]. Figure 4.1 shows a typical scenario of message exchanges in the context of the IEEE 802.11 and 11i. Our focus is on two main steps after (re)association. First, the IEEE 802.1X authentication, where an authentication protocol like TLS [31] operates, is to verify the authenticity of end-to-end principals: the mobile (STA) and the authentication server (AS) via the authenticator (AUTH) (which operates in an AP). In particular, the AUTHs and AS construct an AAA architecture. Successful mutual verification of each identity leads to the derivation of a pair-wise master key (PMK). This key is transferred to the AUTH via a secure tunnel. Second, the STA and AUTH perform the 4-way handshake, exchanging their nonces,

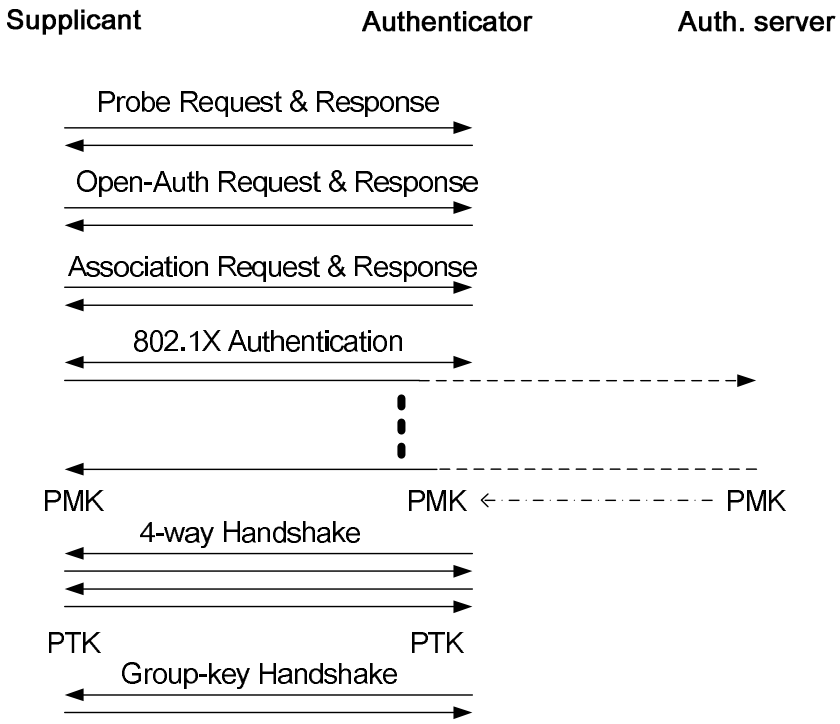


Figure 4.1: Message exchanges in the IEEE 802.11 and .11i systems

so that a pair-wise temporary key (PTK) with which the wireless link will be secured is produced using the PMK as a seed.

The performance of the IEEE 802.11i authentication depends on the efficiency of this authentication protocol. Recent efforts on security associations have been limited to distribution of keys to access points within a domain [58]. For inter-domain handoffs, however, authentication latency is critical to the application QoS.

4.2.2 Cross-domain-related Protocols

There are two protocols: Kerberos that supports the cross-domain authentication and NS that can be effectively extended to do so. We will use the two protocols to comparatively evaluate the throughout of our protocol via simulation. The following are the descriptions

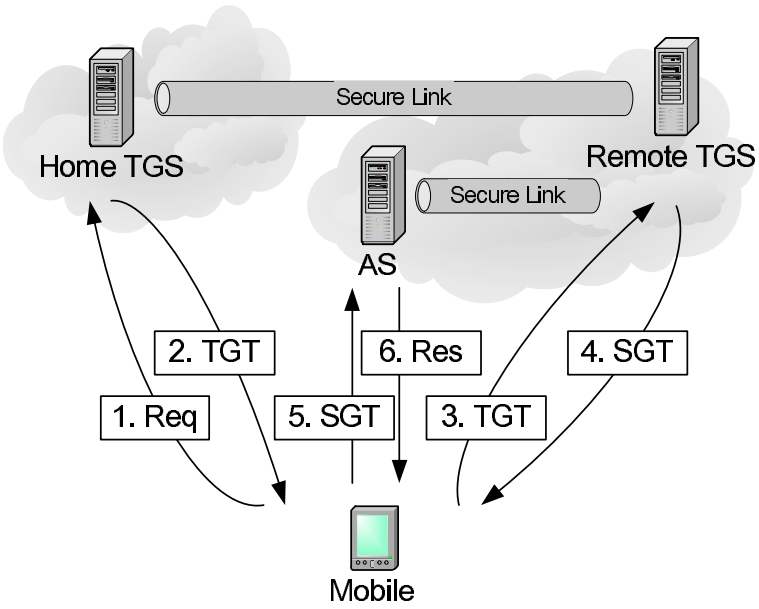


Figure 4.2: Message exchanges for a remote access grant in Kerberos. This sequence is repeated each time the mobile user is bound to a remote TGS.

of the message exchanges of each protocol.

The Kerberos protocol provides cross-domain operations. By establishing inter-domain keys, the administrators of two domains allow the mobile user to receive services in a remote domain. It receives a remote ticket granting ticket (TGT) from the ticket granting server (TGS) in the local domain. It then obtains a service granting ticket (SGT) from the remote TGS in the other domain by using the issued remote TGT. With the SGT containing a secret key, the mobile user and AS can authenticate each other. Figure 4.2 illustrates a sequence of message exchanges for a remote authentication in Kerberos. The link among TGSs is assumed to be secure; a secret key of each TGS is shared to identify itself. In addition to the secure link, the AS has security association with its TGS. The remote TGT issued earlier can be reused to get TGTs in the current domain within a given period of time. However, each time the mobile user moves into a foreign domain, he/she needs to get a remote TGT again by contacting its home TGS.

The NS protocol on which Kerberos is based is not intended to operate over cross-organization boundaries. However, it can support cross-domain authentication with minor modifications, which we call a modified NS protocol (MNS). At first, the original protocol operates, in principle, as can be seen in Figure 4.3. The initiator *A* and its correspondent *B* share secret keys AK and BK with the AS, respectively. In the beginning, *A* obtains two copies of a pair-wise key encrypted with BK and AK by the AS, respectively, during their communication. Then, *A* sends *B* the BK-encrypted pair-wise key along with SK-encrypted AN. AN will be returned in the next message in order for *A* to ensure that *B* with which *A* is communicating is legitimate. *B* also adds BN to the message encrypted by SK, and verifies the decremented BN that *A* sends eventually; those exchanged nonces may be used for key generation for need. We can view *A* as STA, and *B* and AS as foreign and home ASs, respectively. When the foreign AS requires a set of pair-wise keys, the home AS generates and sends a set of multiple different keys. Once receiving them, the foreign AS has no need for contacting the home AS, which may lead message exchanges to be reduced into the 3-way handshake.

4.2.3 Design Requirements

The IEEE 802.11i authentication suggests several requirements that must be preserved to secure WLANs.

- **Requirement 1**: The STA and AS must be able to authenticate each other. Since the STA establishes a wireless link to the AS via anonymous APs, it should be able to identify the AS, so should the AS.

- **Requirement 2**: A successful mutual authentication leads to the derivation of a fresh key for the AS and STA. After the successful mutual authentication, a 256-bit

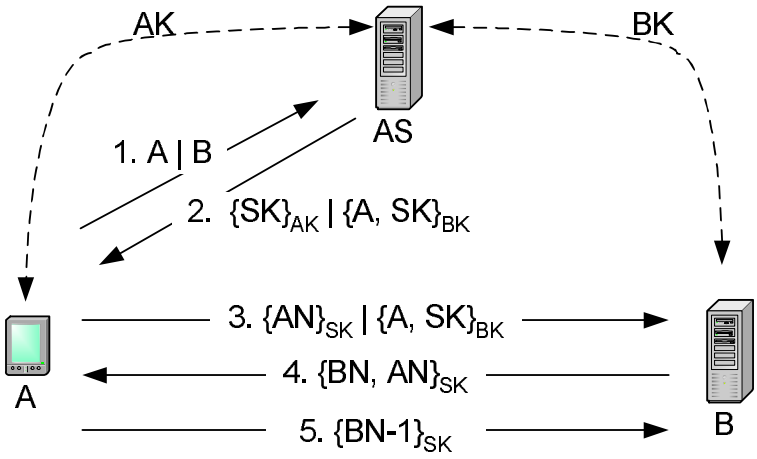


Figure 4.3: An example of message exchanges in symmetry-key-based NS protocol. AK and BK are pre-shared between A and AS, and B and AS, respectively. Elements used to authenticate A or AS in messages 1 and 2 are omitted. AN and BN are *A*'s and *B*'s nonces, respectively.

key (i.e. PMK) is generated by the AS and STA, and is eventually used by the STA and AP. This key must not be reused, and becomes obsolete whenever the STA binds with a new AP.

- **Requirement 3**: Mutual authentication should be strong enough to be protected from any unauthorized reception. It is uneasy to demonstrate the safety of the authentication protocol, but there are theoretical approaches for this purpose. For example, formal verification methods based on model checking and theorem proving, modal logic and modular approach are widely used. We will show a logical proof of MAP using BAN logic in Section 4.4.1. In addition to these requirements, we present the following recommendations for the authentication protocol design to help achieve fast handoffs in WLANs.

- **Recommendation 1**: Minimizing message exchanges during the authentication process helps improve the performance of cross-domain handoffs. We evaluate the

effects of the number of message exchanges.

- **Recommendation 2**: The use of lightweight cryptographic algorithms helps low-power mobile terminals, like personal data assistants, mitigate the performance overhead of computation-intensive cryptographic algorithms.

Based on the above requirements, we will design a protocol supporting cross-domain mobility.

4.2.4 BAN Logic

BAN logic [25] is a modal logic developed for authentication protocol analysis. It presents the proof that a simple logic could be used to describe the beliefs of trustworthy communicating parties. It found redundancies or security flaws in authentication protocols in the literature [24]. BAN logic reasons that the protocol operates as correctly as expected. It is effective to prove the correctness of the authentication mechanisms with logical reasoning.

We introduce the several constructs and logical postulates in BAN logic that will be used for the proof of MAP. Full details of its rules are given in [25]. First, the following are the constructs that we use:

- *P believes X*: *P* believes *X*. In particular, the principal *P* may act as if *X* is true. This construct is essential to the logic.

- *P sees X*: *P* sees *X*. Someone has sent a message containing *X* to *P*, who can read and repeat *X* possibly after doing some decryption.

- *P said X*: *P* once said *X*. The principal *P* at some time sent a message including the statement *X*. It is unknown when the message was sent, but it is known that *P* believed *X* then.

- *P controls X*: the principal P is an authority on X and should be trusted on this matter, e.g. a server is often trusted to generate encryption keys properly. This may be expressed by the assumption that the principals believe that the server has jurisdiction over statements about the generated keys.

- *fresh(X)*: the formula X is *fresh*, i.e. X has not been sent in a message at any time before the current run of the protocol. This is usually true for *nonces* that is randomly generated for use only once.

- $P \overset{K}{\longleftrightarrow} Q$: P and Q may use the shared key K to communicate. It is never disclosed by any principal except for P and Q.

- $P \overset{X}{\Longleftrightarrow} Q$: the formula X is a *secret* known only to P and Q, and possibly to principals trusted by them. Only P and Q may use X to prove their identities to one another.

- $\{X\}_K$: this represents the formula X encrypted under the key K.

- $< X >_Y$: this represents X combined with the formula Y; it is intended that Y be a secret and that its presence prove the identity of whoever utters $< X >_Y$.

Then, we use the following logical postulates in proof.

- The *message-meaning* rules are applied to the interpretation of messages for shared keys

$$\frac{P \text{ believes } Q \overset{K}{\longleftrightarrow} P,\ P \text{ sees } \{X\}_K}{P \text{ believes } Q \text{ said } X}$$

and for shared secrets,

$$\frac{P \text{ believes } Q \overset{Y}{\Longleftrightarrow} P,\ P \text{ sees } < X >_Y}{P \text{ believes } Q \text{ said } X}.$$

- The *nonce-verification* rule represents the check that a message is recent and that the sender still believes in:

$$\frac{P\ believes\ fresh(X),\ P\ believes\ Q\ said\ X}{P\ believes\ Q\ believes\ X}.$$

- The *jurisdiction* rule states that if P believes that Q has jurisdiction over X then P trusts Q on the truth of X:

$$\frac{P\ believes\ Q\ controls\ X,\ P\ believes\ Q\ believes\ X}{P\ believes\ X}.$$

In addition, the HMAC (Hash Message Authentication Code) represented as $MAC(m, K)$, where m and K denote a message and a pair-wise secret key, respectively, is used to verify whether or not the verifiee possesses the same K as the verifier. In other words, only if the generated codes are different, the applied Ks are different. Therefore, $MAC(m, K)$ is interpreted as a unit of the secret $< X >_Y$.

4.3 MAP

In this section, we describe an authentication architecture that extends the AAA architecture to SCR communications, and design MAP. The description of MAP includes the definition of keys and messages, message exchanges, and detailed operations in each functional module.

4.3.1 Architecture

Authentication operations work basically with three entities: STA, AS and AUTH. An STA represents the end user with a WLAN-interface-equipped device. An AS verifies the STA's authenticity and provides each key to secure their wireless link. An AUTH relays authentication traffic between the STA and AS. In addition to dealing with these entities, our protocol solves the cross-domain authentication problem by introducing so-called *security context routers* (SCRs). An SCR is usually placed between multiple AUTHs and an AS. The SCR is logically distinct from the AS in terms of enforcing authentication policy, although both may reside on the same physical machine or the SCR can be integrated into the AS. The SCR functions as follows. After receiving a security context* issued by MAP on the AS, it can perform re-authentication on behalf of the AS. The SCRs are distributed in each domain so that they can reduce authentication latency while the STA roams around the domain. It is assumed that in case of the communication of inter-administration domains they have a security association agreement on roaming and are securely connected to one another by sharing inter-domain keys. This combination is adaptable to the security architecture of the IEEE 802.11i authentication and Wi-Fi Protected Access 2 (WPA2) [5]. The protocol describing how messages are exchanged between the SCRs is presented in Chapter 5. In this chapter, we will give a rough idea of how to exchange messages between SCRs shortly.

Figure 4.4 depicts the MAP architecture. The MAP server module on the AS, which is described in Section 4.3.6, is an end-point authentication protocol that is assumed to securely be connected to the AUTHs via the SCR. The AS used in the architecture is functionally equivalent to the AAA server. The MAP security context module (SC module) in

*Its contents vary with individual protocols. MAP is expected to have a set of authentication value pairs, identity (= mobile Id), validity time, time stamp, mean handoff time, counter and other security information.

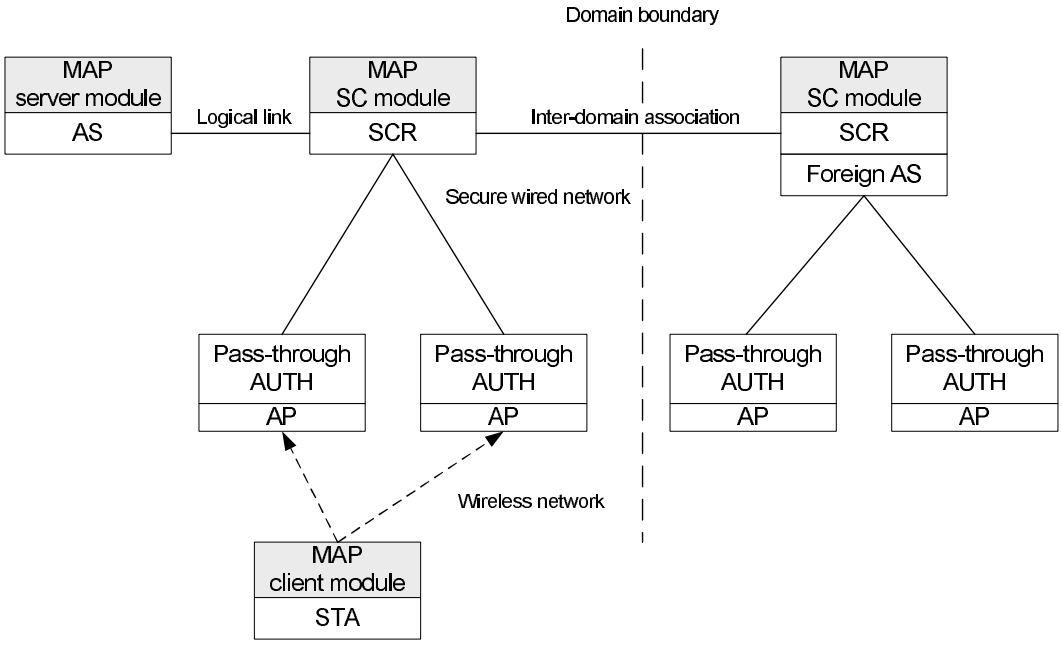


Figure 4.4: Authentication architecture

the SCR, which is described in Section 4.3.6, helps the AS communicate with the other MAP-support AS for cross-domain authentication. The AUTH is an authentication client as a pass-through authenticator. It relays authentication traffic from the STA to the AS, and vice versa. The MAP client module in the STA, which is described in Section 4.3.6, is an end-point authentication party that requests authentication and eventually establishes a secure link with the attached AP.

4.3.2 Communication between SCRs

The SCRs communicate with each other, based on a peer-to-peer manner. There are two ways of transferring security context among the SCRs involved. In case of no security context cached in an SCR with which an STA has just associated, the targeted SCR fetches security context from the original SCR with which the STA associated previously; *reactive*

transfer introduces the latency of fetching security context. On the other hand, the original SCR may somehow forward the targeted SCR(s) security context before the STA is handed off; *proactive transfer* emphasizes the availability of the context ahead of time. On the other hand, estimation of the STA's direction and management of security context can be emphasized, which is referred to as *predictive forwarding* of security context. Their combination yields a tradeoff between storage overhead and latency performance. Elaboration on such issue is part of our future work.

4.3.3 Authentication

The MAP's authentication relies on message authentication code (MAC) algorithms [50]. The MAC values rely on shared symmetric keys, the management of which is uneasy to scale in that two communication parties must somehow exchange the key in a secure way, compared to that of asymmetric-key pairs. However, on the other hand, signing and verifying public keys are very time-consuming; the MAC values are preferred to digital signatures because the MAC computation is two to three orders of magnitude faster. There is a tradeoff between scalability and CPU usage; we chose cost efficiency since it matches our design goal.

4.3.4 Defined Keys

We define three types of keys for different purposes: primary key (PK), domain key (DK) and temporary key (TK). PK is a long-term symmetric key which may be periodically updated and deployed, e.g. online subscription to a service provider or off-line set-up with a purchased card. PK is assumed to have guaranteed protection against disclosure for a sufficiently long period of time. DK is a quasi-primary key in a (sub)domain, which

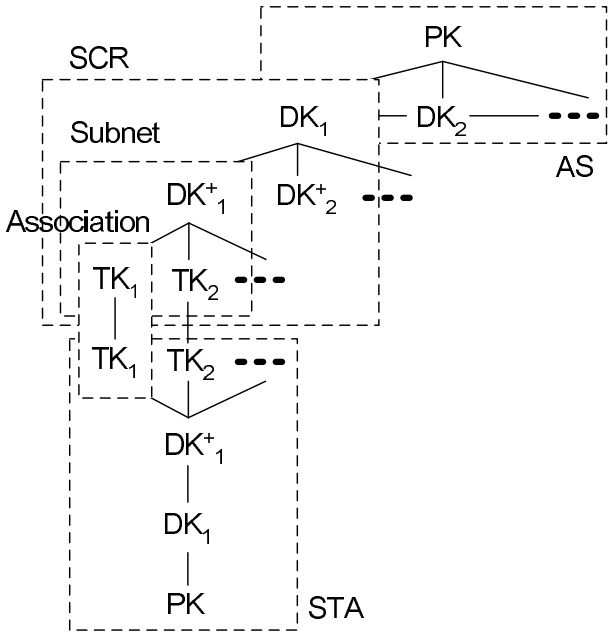


Figure 4.5: Defined keys hierarchy and boundary. An SCR controls a DK derived from the PK. A subnet uses a DK^+ hashed from the DK to generate TK that will be used for each association.

is derived from PK and the previous DK. The STA generates a new DK as it changes a domain; an old DK must be revoked. In addition, DK^+, an *n*-time-hashed DK, is defined for use in a subnet within a domain — if no concept of such subnet is applicable DK^+ is generated from each DK; it plays a role of *loose coupling* of DK and TK. TK that is derived from DK^+ is a link key affiliated with securing a wireless link established between the STA and AP. TK binds with the addresses of two involved physical devices. Therefore, in case of re-associating or changing a binding address, TK is also changed. Figure 4.5 shows a hierarchical derivation and boundary of the defined keys. An association is made of each TK; the disclosure of any TK has no effect on other (re)associations. DK^+ also provides a key-disclosure barrier for relation between TK and DK. AS affects only the generation of DKs.

4.3.5 Defined Messages

We define six types of messages exchanged, with the server, SC, authenticator and client modules interacting with each other during initial and re- authentication in MAP. The first four messages are used during the initial authentication and the last two are used during re-authentication.

- *Auth-req* message, sent by the client module in the STA, triggers a negotiation on authentication and key agreement from scratch.
- *Auth-chal* message, sent by the server module, as a return message, is used for the purpose of challenging the STA, with an encrypted code used for verifying the AS's authenticity to the STA.
- *Chal-res* message, sent by the client module, as a response message, contains a nonce-response encrypted code so that the AS verifies the STA's authenticity.
- *Auth-res* message, sent by the server module, is a reply to a challenge-response message.
- *Reauth-req* message is sent by the client module in the authenticated STA. The SCR captures this message and verifies if the authentication code is legitimate.
- *Reauth-res* message is a reply to the reauthentication-request message including the authentication result.

4.3.6 Message Exchanges

The following is an example of exchanging messages in case of a successful authentication. Only authentication-related information is highlighted in the messages.

M1. STA → AS: *Auth-req*(STAId, SN_i)

The STA sends the AS an authentication request containing its identity (Id) and a fresh nonce. On receipt of the message, the AS fetches credential corresponding to the Id and extracts its key from it.

M2. AS → STA: *Auth-chal*(MAC_{PK}[STAId, SN_i, ASN_j, 'authch'])

The AS uses the STA's nonce to compute a MAC, which protects from a reply attack.

M3. STA → AS: *Chal-res*(MAC_{PK}[STAId, SN_{i+1}, ASN_j, 'authres'])

If the received MAC is matched with the one that the STA generates, the AS is authenticated to the STA. Subsequently, the STA responds to the challenge. Otherwise, the message is silently ignored.

M4. AS → STA: *Auth-res*(ENC_{DK^+}[SN_{i+1}, ASN_{j+1}, AUN_k])

If the STA is successfully authenticated as well, the AS adds a *secret* value, i.e. ASN_{j+1} to a response message. During transfer of the message, the authenticator inserts a newly generated nonce AUN_k that is used to compute a temporary key (TK). Meanwhile, the AS computes and sends a set of authentication value pairs (AVPs) to the SCR.

When the STA re-associates with another AP in (another) subnet, the following messages are exchanged.

M5. STA → SCR: *Reauth-req*(MAC_{PK}[STAId, SN_{i+2}, ASN_{j+1}, DK_i, 'reauth'])

The STA computes a MAC, using the secret value obtained in the previous round of authentication. The SCR can verify if the STA holds the same nonce.

M6. SCR → STA: *Reauth-res*(ENC_{DK^+}[SN_{i+2}, ASN_{j+2}, AUN_{k+1}])

If the STA is authenticated successfully, the SCR adds another nonce for the next challenge in the message. The STA can authenticate the SCR by verifying if the nonce is identical of the one that it sent previously.

In the subsequent section, we describe in details how MACs and hierarchical keys are computed and used in each module.

MAP Server Module

The server module handles two types of incoming messages (i.e. *auth-req* and *chal-res*) that are related only to authentication from scratch. The following is the description of the pseudo code of the server module.

var: $sn_{1..n}$, $cn_{1..n} := 0$; %Server and client nonce queues are initialized.

for all i: *auth-req* of Id_i in buffer **do**

sn_i=refresh(sn_i); %A fresh nonce is generated.

send *auth-chal*: $sn_i \mid \mathrm{MAC}_{PK_i}(\mathrm{Id}_i, cn', sn_i,$ "*authch*");

cn_i=cn'; %Client nonce from the message is buffered.

end for

for all i: *chal-res* of Id_i in buffer **do**

$DK_{i,j-1} = \mathrm{PRF}(PK_i, cn_i, sn_i)$; %$cn_i$ is obtained from $auth-req$.

if $\mathrm{MAC}_{PK_i}(\mathrm{Id}_i, cn', sn_i,$ "*authres*") && $\mathrm{MIC}_{DK_{i,j-1}}$ verified

%cn' is obtained from *chal-res*.

sn_i=refresh(sn_i);

DK^+=$H^{\alpha_i}(DK_{i,j-1})$

send *auth-res*: $sn_i \mid cn' \mid DK^+$; %DK^+ is transferred to the authenticator.

make SC_i:

for $e = 1..n$ **do**

$\text{MAC}_{PK_i}(\text{Id}_i, sn_i, DK_{i,j-1}$, "*reauth*");

$DK_{i,j}$=PRF(PK_i, $DK_{i,j-1}$, sn_i);

AVP_e:(Id_i, sn_i, MAC, $DK_{i,j}$) $\in \bigcup_{1..e-1}$ AVP;

sn_i=refresh(sn_i);

end for

end if

end for

A MAC, including client nonce cn' from the received message and server nonce sn_i, is computed and sent to the STA of Id_i. The MAC allows the STA to verify the AS's authenticity. DK is computed by calculating an n-bit key generating pseudo random function (PRF) — in most cases n=128 is sufficient — with PK and the previously-exchanged nonces. An MIC provides a means of verifying authenticity once the associated MAC is verified successfully. A hashed domain key, DK^+, is generated by applying α times a cryptographic one-way function H, equivalently $H^{\alpha}(x) = H^{\alpha-1}H(x)$ and $H^0(x) = x$. The α value is a sync-one shared between the STA and the AS/SCR. DK^+ allows DK to be hidden from the authenticators. After the message exchanges, the server module creates the STA's security context that is composed primarily of the set of AVPs. It is then transferred to the corresponding SCR. The AVPs enable the SCR to conduct the re-authentication and re-keying process on behalf of the AS.

MAP SC Module

The SC module handles an incoming message (i.e. *reauth-req*) and an outgoing message (i.e. *reauth-res*) which are related to re-authentication. In particular, this module can be

implemented, combined with the server module. The following is the description of the pseudo code of the SC module.

for all i: *reauth-req* of Id_i in buffer **do**
AVP_l=(Id_i, sn_i, MAC, $DK_{i,j}$) $\leftarrow \bigcup_{l..n}$ AVP; %`Select one of AVPs.`
if MAC && $\mathrm{MIC}_{DK_{i,j}}$ verified **do** %`The integrity of the message is checked.`
send *reauth-res*: $cn'|sn_i|H^{\alpha_i}(DK_{i,j})$; %`DK`$^+$ `is derived by` α`-time hashing.`
end if
end for

The SC module first retrieves one of AVPs from the security context corresponding to Id_i and then verifies MAC≠MAC′ or $\mathrm{MIC}_{DK_{i,j}}$(*reauth-req*)≠MIC′. If they are matched correctly, it computes DK^+ and sends the authenticator it along with the exchanged and retrieved nonces. If DK is not allowed to be reused, the AVP is dethroned when it is notified somehow that the STA of Id_i de-associates with the current AP. If no more AVP exists, the re-authentication request is forwarded to the AS which will, in turn, handle the request from scratch. Note that the SC module does not possess any PK.

Authenticator

A primary role of this module is to relay incoming messages. It also computes an TK with which the STA and AP establish a secure link after a successful authentication.

var: an; %`This is an authenticator nonce.`
if *auth-req* | *auth-chal* | *chal-res* | *reauth-req* received
relay it;

end if

if *auth-res* | *reauth-res* received

if success in authentication %`This is determined by AS/SCR.`

an=refresh(*an*); % `A new` *an* `is used to generate TK.`

send *auth-res*: $ENC_{DK^+}[sn' \mid an \mid cn']$;

%`DK`$^+$ `and` *cn*′ `are obtained from the message.`

TK=PRF(DK^+, $Addr_{STA} \mid Addr_{AP}$, $an \mid cn'$);

end if

end if

Authenticator is beyond access to DK; DK^+ received from the AS/SCR is used to compute TK by calculating a PRF — the key-size varies with cryptographic protocols to be used for securing a wireless link, yet it is either 256 or 512 bits. TK binds with media access control addresses of the STA and AP; de-association revokes TK and a new TK must be recomputed.

MAP Client Module

The client module incurs an authentication request message (e.g. *auth-req* or *reauth-req*) when the STA (re)associates with an AP. It also handles incoming messages (i.e. *auth-chal*, *auth-res* and *reauth-res*) and outgoing messages (i.e. *chal-res*).

var: *secret* := 0, *cn*; %*cn* `is a client nonce.`

if (re)associated

cn=Refresh(*cn*);

if !*secret* %`In case of authentication from scratch`

send *auth-req*: Id | *cn*;

else %`In case of the previous successful authentication`

DK_i=PRF(PK, DK_{i-1}, *secret*);

send *reauth-req*: Id|MAC_{PK}(Id, *secret*, DK_{i-1}, "*reauth*")|*cn*|MIC_{DK_i};

end if

end if

if *auth-chal* received

if MAC_{PK}(Id, *cn*, *sn*′, "*authch*") verified %`It authenticates AS.`

DK_{i-1}=PRF(PK, *cn*, *sn*′); %*sn*′ `is obtained from` *auth-chal*

cn=Refresh(*cn*);

send *chal-res*: Id|*cn*|MAC_{PK}(Id, *cn*, *sn*′, "*authres*")|$\text{MIC}_{DK_{i-1}}$;

end if

end if

if *auth-res* | *reauth-res* received

DK^+=H^{α}(DK_{i-1} or DK_i);

DEC_{DK^+}[ENC[*sn*′ | *an*′ | *cn*′]];

if *cn*==*cn*′ %`It authenticates AS.`

secret=*sn*′; %*sn*′ `is stored as secret`

TK = PRF(DK^+, $Addr_{STA}$ | $Addr_{AP}$, *an*′ | *cn*);

%*cn* `is obtained from the previous message.`

end if

end if

It retains the *secret* value provided by the AS after completion of the previous successful authentication. Confidentiality of the secret is guaranteed since it is transferred in

ciphertext. The secret determines whether the authentication process is conducted from scratch. The α value is matched to that of the AS/SCR.

4.4 Security Considerations

In this section, first, using BAN logic, we show the logical proof that MAP performs its authentication mechanism correctly as it is expected, and then examine security threats to our protocol.

4.4.1 Protocol Analysis

The analysis procedure works as follows. First, we translate the original protocol into the idealized one and then make assumptions about the initial state. Finally, we make logical formulas as assertions and apply the logical postulates to the assumptions and assertions to arrive at conclusions.

Translation; we extract the encrypted forms of messages from MAP communications as follows:

M1. B $\rightarrow$ A: $< N_a, N_b >_{PK}$

M2. A $\rightarrow$ B: $< N_b, N_a >_{PK}$

M3. B $\rightarrow$ A: $\{N_{b'}, N_{a'}\}_{DK}$

M4. A $\rightarrow$ B: $< N_{b'}, A \overset{DK}{\longleftrightarrow} B >_{PK}$

M5. B $\rightarrow$ A: $\{N_{b''}, N_{a''}\}_{K_{ab'}}$

We have STA and SCR, referred to as A and B, respectively — the functionality of AS and AUTH is integrated into SCR for simplicity; DK^+ is identical of DK. We also omit communication in clear-text. There is a slight difference by representing $(N_a \oplus N_b)$ as (N_a, N_b), which is acceptable since this means that N_a and N_b were uttered at the same time and their XOR-ed value is straightforwardly obtained.

For authentication, each party verifies the MAC which requires the nonces generated by itself and the other. That is, the correct MAC can only be generated with the fresh nonces from the two. Thus, authentication between A and B might be deemed complete if each of the two believes that the other has recently sent the nonce, and proving sound mutual authentication is sufficiently satisfied by deriving the facts:

$$A \textit{ believes } B \textit{ believes } N_a \text{ and } B \textit{ believes } A \textit{ believes } N_b$$

for initial authentication and

$$A \textit{ believes } B \textit{ believes} N_{a''} \text{ and } B \textit{ believes } A \textit{ believes } N_{b'}$$

for re-authentication.

Making assumptions; we then write the following assumptions:

(1) A *believes* $A \overset{PK}{\Longleftrightarrow} B$, (2) B *believes* $A \overset{PK}{\Longleftrightarrow} B$,

(3) A *believes* $A \overset{DK}{\longleftrightarrow} B$, (4) B *believes* $A \overset{DK}{\longleftrightarrow} B$,

(5) A *believes* $A \overset{DK'}{\longleftrightarrow} B$, (6) B *believes* $A \overset{DK'}{\longleftrightarrow} B$,

(7) A *believes* $fresh(N_a)$, (8) B *believes* $fresh(N_b)$,

(9) A *believes* $fresh(N_{a'})$, (10) B *believes* $fresh(N_{b'})$,

(11) A *believes* $fresh(N_{a''})$, (12) B *believes* $fresh(N_{b''})$,

(13) *A believes* $fresh(N_{b'})$, (14) *A believes* $fresh(N_{b''})$,

(15) *A believes B controls* $N_{b'}$,

(16) *A believes B controls* $N_{b''}$.

Assumptions (1) and (2) are made from the fact that A and B initially share a secret, PK. Assumptions (3), (4), (5) and (6) are derived from the fact that only A and B can generate a shared key only if the sound authentication is achieved. Assumptions (7) to (12) state that A and B believe that the nonces generated by themselves are fresh; freshness of nonces holds by verification of MAC and MIC associated with the nonces. The nonces, $N_{b'}$ and $N_{b''}$, also play a role of secrets since they are transferred with proper encryption. Thus, A can believe that B has generated the nonces that was not used in the past,which leads to Assumptions (13) and (14), and also (15) and (16), indicating that A trusts B to generate the secret.

Reasoning; we analyze the idealized version of MAP by applying the logical postulates presented in Section 4.2.4 to the assumptions.

A receives Message M1. The annotation rule yields that A *sees* $< N_a, N_b >_{PK}$ holds afterward. With the hypothesis of (1), the message-meaning rule for shared secrets applies and yields A *believes* B *said* (N_a, N_b). Breaking conjunctions produces A *believes* B *said* N_a. With the hypothesis of (7), we apply the nonce-verification rule and yield A *believes* B *believes* N_a. On the other hand, B receives Message M2 and the following result is obtained in the same way as that of Message M1, via the message-meaning and nonce-verification rules with hypotheses (2) and (8), respectively, B *sees* $< N_b, N_a >_{PK}$ and B *believes* A *believes* N_b. This concludes the analysis of Message M2. The analysis of Messages M1 and M2 confirms that MAP performs mutual authentication successfully.

A receives Message M3 and the annotation rule yields that *A sees* $\{N_{b'}, N_{a'}\}_{DK}$ holds thereafter. The message-meaning rule for shared keys with the hypothesis of (3) via breaking conjunctions yields: *A believes B said* $N_{b'}$, and *A believes B said* $N_{a'}$. Taking the former, with hypotheses (13) and (15), the nonce-verification and jurisdiction rules apply and yield *A believes B believes* $N_{b'}$, and *A believes* $N_{b'}$, respectively. Taking the latter, the nonce-verification rule with hypothesis (9) yields *A believes B believes* $N_{a'}$. This concludes the analysis of Message M3. This message may appear redundant since authentication is completed from Message M1, but it is essential not because it is for authentication, but because it is for transmission of a secret, nonce $N_{b'}$.

B receives Message M4 and the annotation rule yields that *B sees* $< N_{b'}, A$ and $\overset{DK}{\longleftrightarrow} B >_{PK}$ holds thereafter. By applying the message-meaning rule for the secrets with (2) via breaking conjunctions, we obtain: *B believes A said* $(N_{b'}, A \overset{DK}{\longleftrightarrow} B)$ and *B believes A said* $N_{b'}$. The nonce-verification rule with hypothesis (10) yields that *B believes A believes* $N_{b'}$. On the other hand, *A* receives Message M5 and the annotation rule yields that *B sees* $\{N_{b''}, N_{a''}\}_{DK'}$ holds thereafter. By applying the message-meaning rule for the shared keys with hypothesis (6) via breaking conjunctions, we obtain *A believes B said* $N_{b''}$ and *A believes B said* $N_{a''}$. Taking the former, the nonce-verification and jurisdiction rules with (14) and (16) yield *A believes B believes* $N_{b''}$, and *A believes* $N_{b''}$, respectively. Taking the latter, nonce-verification with (11) yields that *A believes B believes* $N_{a''}$. The analysis of Messages M4 and M5 confirms that MAP also achieves mutual re-authentication.

4.4.2 Possible Attacks

Key recovery attack: This relies on finding the key K itself from a number of message–MAC pairs. Ideally, any attack allowing key recovery requires about 2^k operations where k

is the length of K. The adversary tries all possible keys with a small number of message–MAC pairs available. Choosing a sufficiently long key is a simple way to thwart a key search. Another possible attack is to choose an arbitrary fraudulent message and append a randomly-chosen MAC value. Ideally, the probability that this MAC value is correct is equal to $1/2^m$, where m is the number of bits in the MAC value. Repeated trials can increase the corresponding expected value, but a good implementation will be alert to repeated MAC verification errors.

Forgery attack: This attack relies on prediction of $\mathrm{MAC}_K(x)$ for a message x without initial knowledge of K. For an input pair (x, x') with $\mathrm{MAC}_K(x)=g(H)$ and $\mathrm{MAC}_K(x')=g(H')$, where g denotes the output transformation and H is a chaining variable, a collision occurs if $\mathrm{MAC}_K(x) = \mathrm{MAC}_K(x')$. Its feasibility depends on an n-bit chaining variable and the MAC result. Given g that is a permutation, a collision can be found using an expected number of $\sqrt{2} \cdot 2^{n/2}$ known text-MAC pairs of at least two divided blocks each. A simple way to counter this attack is to ensure that each sequence number at the beginning of every message is used only once within the lifetime of the key.

Impersonating attack: Note that the AUTH, SCR and AS maintain a security association with each other. Therefore, neither of them can be used to impersonate the other. Instead, this attack occurs between the STA and AUTH, which causes an authentication failure or misconduct of the principals. Oracle-based impersonating attacks are that the attacker exploits one of principals as an oracle to obtain cryptographic messages in a session since it has no knowledge of K. The attacker applies the obtained messages to the other principal party in another session. For example, it runs a session with an AUTH to obtain a MAC value, impersonating a legitimate STA. It runs another session with an STA and

exploits the MAC value on the STA, impersonating the legitimate AUTH. This attack can be countered by exchanging nonce with each other and using a sequence counter.

4.5 Performance Evaluation

We evaluate the efficiency of MAP via experimentation and simulation, contrasting it with other protocols. We first describe simulation methodology and model and then analyze the MAP's performance benefits via the simulation results and in comparison with other protocols. Finally, we discuss the storage overhead caused by security-context transfer.

4.5.1 Simulation Methodology

The probe phase, discovering the next AP in WLAN handoffs, takes a large latency (ranging from 50*ms* to 350*ms*), depending on different vendors [56]. Even if the recent effort in [75] to reduce latency by 84%, the large variance is an obstacle to highlight the effectiveness of our protocol on a real testbed. We therefore use Matlab-based simulation, relying on experimental data. We assume that network traffic is stable with small variations, e.g. the latency of establishing a (re)association with an AP including the probe phase is 30*ms* with 3% jitter, and the round-trip time (RTT) between two communicating servers across a domain is about 20*ms* with 4% jitter. In addition, the RTT between the AP and SCR/AS is less than 3*ms*. We use these values throughout the simulation. In cryptographic computations, we conducted an experiment using three machines: Linux v.2.4.19 iPAQ 206MHz ARM processor with 64 megabyte memory (iPAQ), Linux v.2.4.2 Laptop Mobile Pentium 366MHz processor with 128 megabyte memory (MP2) and Linux v.2.4.23 Desktop Intel Xeon 3Ghz bi-processor with 2GB memory (Xeon). We compiled crypto libraries [30] in gcc v.3.3 with an option of Level-1 optimization.

Table 4.1: Throughput of hash/symmetric and asymmetric algorithms (in Megabit per second)

Alg.\Pow.	iPAQ	MP2	Xeon
SHA-1	15.8 Mbps	18 Mbps	104.9 Mbps
SHA-256	3.4 Mbps	9 Mbps	64.0 Mbps
SHA-512	0.2 Mbps	4.3 Mbps	24.8 Mbps
MD5	15.8 Mbps	41 Mbps	290.9 Mbps
AES-128	2.7 Mbps	10 Mbps	80 Mbps
RSA enc.	15.1 Kbps	138.9 Kbps	625 Kbps
RSA dec.	0.9 Kbps	4.6 Kbps	21.6 Kbps
RSA sig.	0.9 Kbps	4.4 Kbps	21.2 Kbps
RSA ver.	15.1 Kbps	138.9 Kbps	625 Kbps

Table 4.1 shows the computation throughput of symmetric-key and public-key algorithms, respectively. With these measurement data, we numerically calculate the time to perform each authentication protocol while ignoring the overhead of running applications for simplicity.

4.5.2 The Simulation Model

Figure 4.6 shows the simulation model we used. Each AS constructs a domain consisting of an SCR and several APs. The SCR and AS may reside on the same machine as mentioned before.

Handoff Pattern

The handoff pattern for STAs is basically random; the STAs cross the boundary after hopping a random number of times. Random pattern is sufficient to evaluate the overall efficiency performance. Nevertheless, to notice the comparative effectiveness of our protocol, we additionally set a regular handoff pattern; after association in the home domain, STAs hop three times and then cross a domain boundary. In a visited domain, every five hops

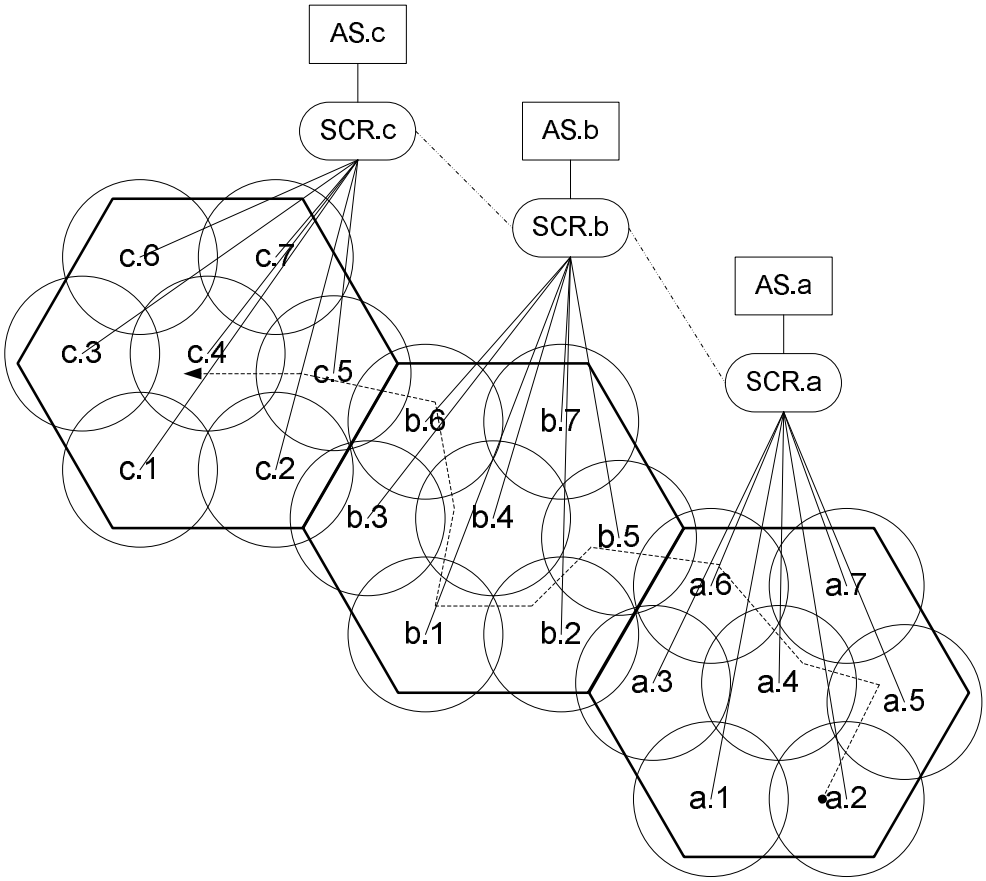


Figure 4.6: The simulation model for inter-domain handoffs. A circle and hexagon indicate an AP's radio coverage range and a domain, respectively. Each SCR controls its domain and is securely connected with its neighbor. An STA initially associates with a.2 and move around in its local domain (from a.5 to a.6 via a.4). It crosses Domain b and finally associates with c.4 in Domain c.

they traverse the domain.

SCR Configuration

Whether or not the "visitor" can use storage resources in a domain affects the performance of its handoffs. There can be three system configurations according to the storage availability in the SCR of the visited domain. First, if only *relaying security context* is allowed, the authentication process takes place in the AS/SCR of the home domain. The SCR in the visited domain serves as a relay agent. Second, if *caching security context* is allowed, the foreign SCR serves as a proxy authentication server. In this case, security context is transferred and stored in the visited domain, which enables avoiding contact with the home server. Third, if *pre-caching security context* is allowed somehow, i.e. security context is transferred to the foreign SCR before the STA arrives, then the latency of fetching security context from the home server/SCR can be eliminated. We will evaluate the caching effect via simulation.

4.5.3 The Simulation Results

MAP performs an optimized re-authentication procedure based on the security context generated after the initial authentication. It allows one to (1) consolidate the re-authentication procedure (with two-message exchanges, the mutual authentication is completed) and (2) avoiding contact with the home server from the visited domain. Figure 4.7 clearly shows that from re-authentication, authentication latency dramatically drops by up to 45% thanks to (1). As a regular handoff pattern, after three hops in the local domain (the first handoff corresponds to the initial authentication in the figure), the STA crosses the domain boundary at every 5 handoffs, which triggers the foreign SCR to request the security context from

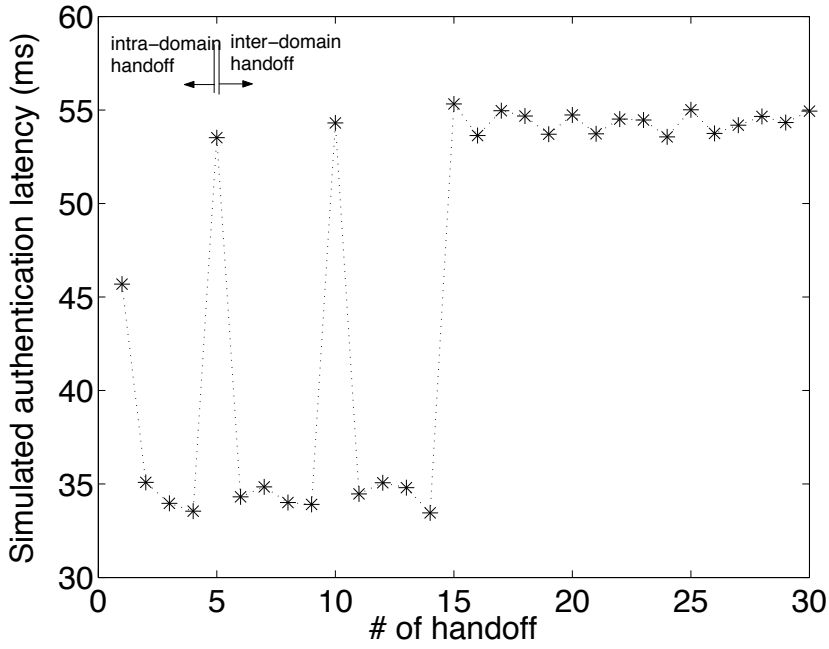


Figure 4.7: Authentication latency variations in different configurations of foreign servers

the home server. As a result, latency increases in proportion to the RTT between the end-to-end points of two domains. Even if the STA roams in the foreign domain, it shows the same latency performance as in the home domain thanks to (2). In this case, the SCR in the foreign domain supports caching security context. After the 15-th handoff in the figure, the cross-domain authentication encounters the case of relaying security context in the SCR of the visited domain, which triggers the authentication procedure to be performed in contact with the home server for each hop in the visited domain.

Figure 4.8 shows the results with a random handoff pattern, illustrating the cumulative distributions of authentication latency for three cases supporting SCR. The figure shows the effect of pre-caching and caching security context to achieve more improvements in time efficiency than just relaying security context which is characteristic of the legacy protocols that are unable to generate security context. For example, more than 70% and 80% of

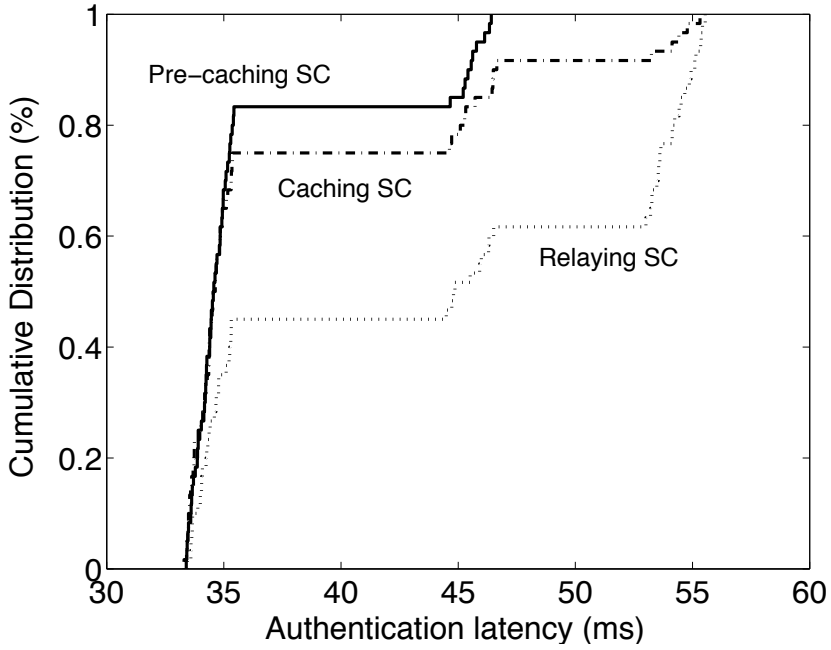


Figure 4.8: Cumulative distributions of authentication latency under each different configuration. SC stands for security context.

authentication processes in the cases of caching and pre-caching security context, respectively, take less than 36*ms*.

We evaluated the increase in storage availability via the number of authentication requests with a random handoff pattern. Figure 4.9 shows that the higher inter-domain handoff frequency the home SCR has, the higher its storage availability. The x-axis is the ratio of authentication request queries in inter-domain handoffs to the total number of queries, and the y-axis is the ratio of the network traffic in the foreign SCRs. Let AQ_r denote the foreign server's overhead and AQ_l denote the home server's overhead. Then, the ratio of the gain in storage availability with MAP to the overall overhead is expressed as, $1 - AQ_l/(AQ_l + AQ_r)$ which grows as the frequency of the inter-domain handoffs increases.

As shown in Figure 4.10 that plots the results with a random handoff pattern, the performance in authentication efficiency (caching allowed) improves up to 53% over a legacy

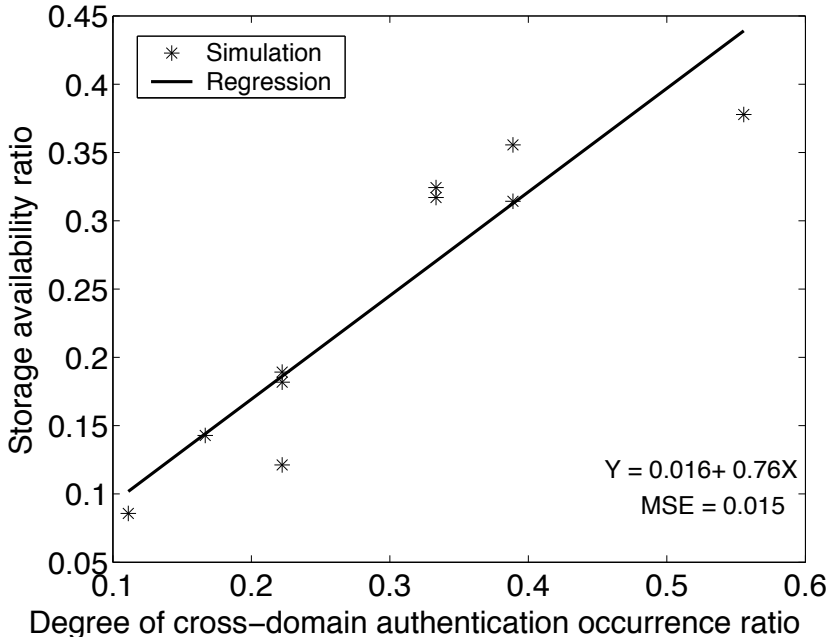


Figure 4.9: System storage availability affected by the inter-domain handoff authentication occurrence ratio. MSE is Mean Squared Error of the above regression function.

method (relaying allowed) until the end-to-end domain distance continues to increase up to RTT=100*ms*. In case of security context pre-cached in the visiting domain, MAP makes a 10% additional improvement with RTT=100*ms*. Therefore, the effectiveness of MAP increases dramatically as the distance gets larger.

4.5.4 Comparison with Other Protocols

Figure 4.11 shows the cumulative CPU usage (represented in millisecond) cryptographic primitives of required in ten consecutive times of authentication in symmetry-key-based protocols including MAP, MNS and Kerberos, and public-key-based protocols including PNS and TLS. We chose one-way hash functions (i.e. MD5 [71], SHA [6]) and block ciphers (i.e. AES [8]) for symmetric-key protocols, and RSA [40] 1024-bit modulus for the public-key protocols. The symmetric-key protocols are shown to be two orders of

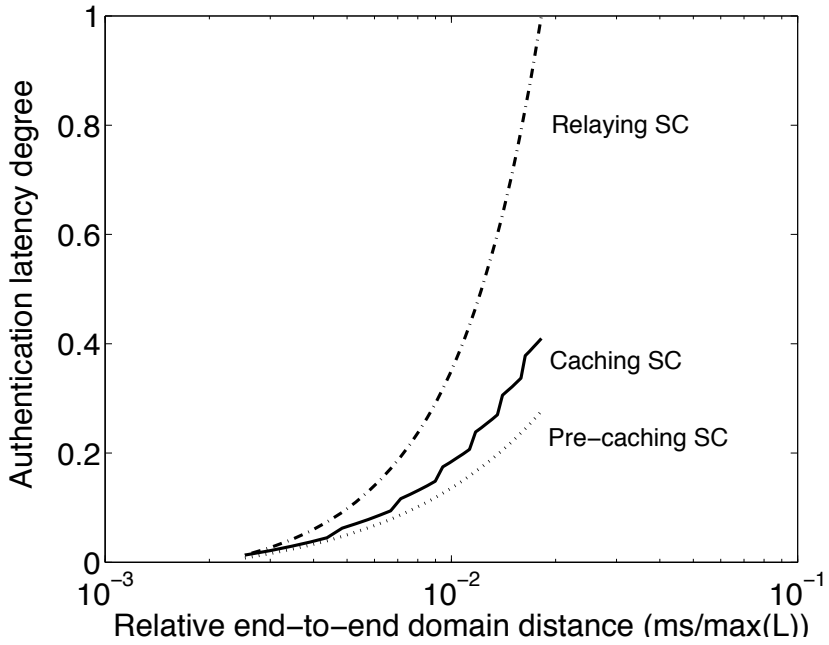


Figure 4.10: End-to-end domain distance vs. authentication latency. The distance is scaled down at a rate of the maximum authentication latency (max(L)). SC stands for security context.

magnitude faster thanks to the inherent advantage over modulo operations. MAP is faster than the MNS and Kerberos protocols, respectively, by 12.6% and 21.5% CPU usage gains. This is a considerable impact on the performance gain in view of millions of runs for authentication in a single server.

Regarding the number of message exchanges, MAP achieves the cross-domain authentication only with 2-way handshake, the cost of which is minimal, compared to MNS and Kerberos requiring 3-way and 4-way handshakes, respectively. This contributes to the further enhancement of latency performance. Figure 4.12 shows the comparison of authentication latency of MAP with that of the MNS and Kerberos protocols while mobile devices are hopping with a regular pattern. MAP outperforms the others in both inter- and intra-domain roaming. It accounts for 74% of cross-domain authentication latency of Kerberos and 85% of that of MNS. It reduces intra-domain authentication latency by 5% for

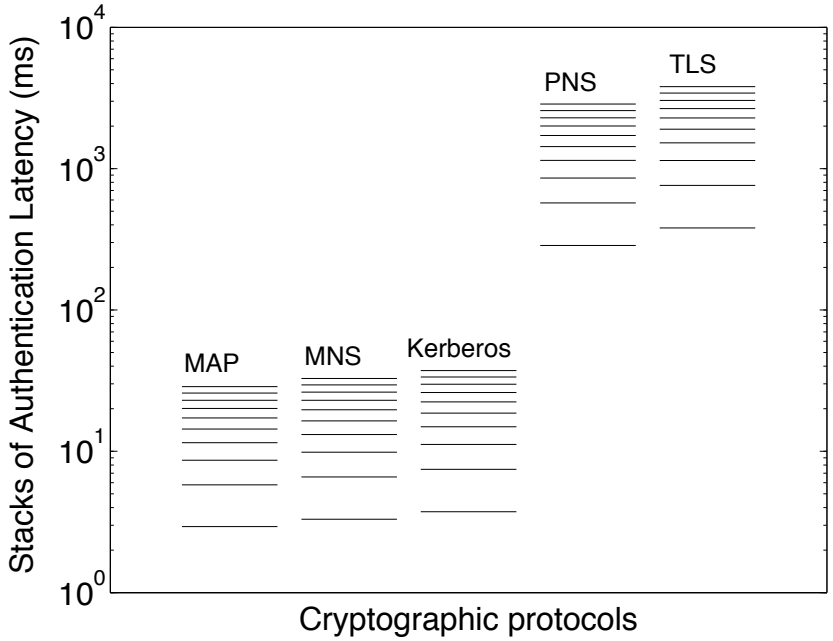


Figure 4.11: CPU utilization. Ten consecutive times of authentication. MNS and PNS stand for modified symmetry-key-based and public-key-based Needham Schroeder protocols, respectively. TLS is Transport Layer Security protocol.

Kerberos and 7% for MNS.

4.5.5 Storage Overhead

Security context is transferred and stored in a foreign server (SCR) for cross-domain authentication. It consists mainly of a set of AVPs each of which is composed of nonce (128 bits), MAC (128 bits), DK (128 or 256 bits) and Identity (about 320 bits). In addition, a value (of 40 bits) may be reserved for security context validity and other information. The security context can be of $64 \cdot n + 45$ bytes where n is the number of AVPs. Approximately, given a 1 kilobyte security context per STA, manipulating one million STAs requires 1 gigabyte storage capacity, which is usually a small overhead to the server system.

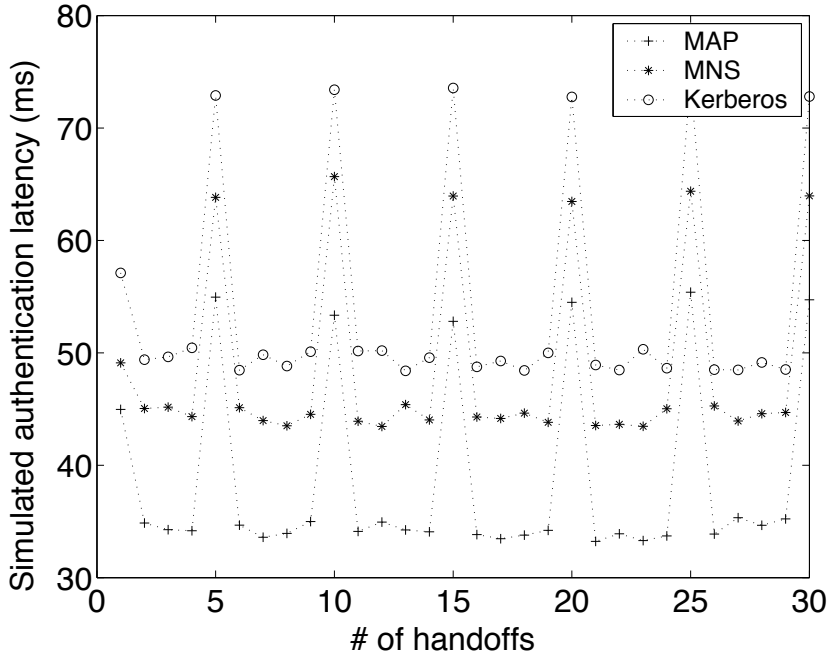


Figure 4.12: Latency comparison of MAP with MNS and Kerberos. Fetching security context while the mobile device crosses the boundary increases authentication latency.

4.6 Related Work

There have been several studies on how to achieve fast handoffs and enhance the performance of authentication mechanisms, including WLAN protocols.

Michra *et al.* [57, 58] presented a keys distributing method by means of proactive context caching. The idea of proactive caching is for an AP to broadcast its cached context to its neighbor APs in advance by using neighbor graphs and IAPP. However, this method is limited to intra-domain handoffs since APs are required to be functionally identical. Pack *et al.* [65] presented a pre-authentication method that skips the IEEE 802.1X authentication phase by distributing the key to a certain number of selected APs and computing the likelihood based on the analysis of past network behavior. Bargh *et al.* [19] presented the applicability of the pre-authentication method for inter-domain handoffs. However, a

pre-authentication method creates a higher risk of compromising security.

Wong *et al.* [80] proposed a hybrid protocol based on a certificate containing a symmetric key signed with a public key which is suitable for wireless communications. An asymmetric method for wireless communications presented in [38] uses Diffie-Hellman key exchange combined with Schnorr signatures. In addition, there are several legacy authentication protocols [25, 31, 64, 74, 81, 84] for the general purpose in the literature.

There are several approaches to analyzing the security of authentication protocols. One is the formal methods that model and verify the protocol using specification languages and verification tools [55]. It consists of model checking and theorem-proving methods. Application examples [37, 53, 54, 59] demonstrated the feasibility of formally verifying the authentication protocols with general-purpose verification tools. Also proposed in [14, 20, 79] are modular approaches aiming to establish a sound formalization and a security analysis for the authentication problem.

4.7 Conclusions

The cross-domain authentication requires retrieval of security context from the server of the previously-visited or home domain. Contacting a remote server may increase authentication latency significantly. the longer the end-to-end distance, the larger the latency reduction. The longer the end-to-end distance, the larger the latency reduction. If security context is allowed to be pre-cached/transferred before the mobile device arrives, latency can be reduced significantly. In this chapter we designed and evaluated a mobility-adjusted authentication protocol, MAP, by leveraging symmetric-key cryptography for cross-domain authentication and key generation. MAP can be configured to make tradeoffs between performance and storage usage. MAP introduces three concepts to the cross-domain authenti-

cation: (1) a re-authentication mechanism based on a 2-way handshake; (2) the temporary-key generation of the IEEE 802.11i authentication; and (3) security context eliminating the need to contact a remote server. MAP performs best in cases of long end-to-end domain distances and high cross-domain authentication traffic.

Chapter 5

Security Context Router

5.1 Introduction

Rapidly converging inter-wireless technologies enable mobile users carrying multimedia-access devices to roam seamlessly around heterogeneous networks. One of the key access-media-independent handoff services is the cross-domain authentication that consists of security protocols and security-context management. Despite the fundamental importance of security context management, how to achieve efficient transfer, consistency, and low storage overhead of the security context is not well understood.

Numerous studies of location tracking and resource allocation for QoS-sensitive applications have been done on intra-domain handoffs in cellular networks. Tracking a mobile's trajectory based on a hierarchical mobility model (i.e. accounting for local and global mobility) improves the accuracy of location prediction [51]. Alternatively, one may compute the handoff probability based on an aggregate history of handoffs observed in each cell [29]. In contrast, only limited work has been done on cross-domain handoffs; whether or not to comply with the same administrative/operational policy over the entire network

distinguishes generic multi-tier boundary handoffs. A boundary location area that covers a region of network boundary and a boundary location register that is a database cache for the handoff information of mobile users, are created for location update and paging purposes between two tiers [15]. This method requires the boundary location registers to be deployed and maintained in cooperation with adjacent end-systems.

Securing a handoff takes a significant amount of time, especially due to its requirement of contacting the remote authentication server or re-establishing a connection from scratch. In Chapter 4, we presented a solution in which the authentication server in the home domain creates a *security context* after successfully authenticating a mobile user, then transfers it to the mobile's target access network so as to eliminate the need for contacting the home server or re-establishing the connection. Security context for securing handoffs needs to be "propagated" as the mobile user visits different *domains*.

We propose a security context router (SCR) that controls (part of) each domain. A group of collaborating SCRs form an overlay network and operate in a manner that is geographically independent of access networks. Given an authentication request, the SCR in control of the domain which a mobile node is visiting, contacts the original SCR of the domain in which the mobile resided previously, and fetches the corresponding security context; this is the *reactive* transfer of the security context. However, it incurs an unpredictably long delay. One may make the security context available in the targeted domain before the mobile arrives; this is called the *proactive* caching of the security context. If the original SCR estimates the mobile will cross the domain boundary, it transfers the replicas of security context to its neighboring domains in the SCR's vicinity. However, care needs to be taken for replication in terms of storage overhead, consistency, and scalability.

Our goal is to provide a method for efficient, *predictive forwarding* of the security context via an overlay network. Our approach differs from most existing prediction approaches

in that it is free of geographical dependency, while the others rely on geographical dependency on wireless/wired access networks. Two key features of SCR are approximate pattern matching and statistical estimation. First, as the mobile experiences handoffs, *Time-history Vectors of Angles* (TVAs) are constructed by capturing the positions of access nodes (including access points or base stations) with which the mobile has been associated, calculating the variations of angles, and observing the fact that a node's mobility reflects its directional pattern. Second, after classifying TVAs into a pattern, we analyze associations of the pattern with target SCRs. Such associations assume a distribution, so that we can determine a confidence interval in forwarding the security context to the SCRs that fall within such an interval. These two features enable accurate prediction of forwarding and significant reduction in storage overhead.

The remainder of this chapter is structured as follows. Section 5.2 describes the overall system on which SCRs are built. We first specify the attributes of security context, and introduce an SCR. We then describe the access networks in which mobiles move around and the overlay network formed by properly linking SCRs. Section 5.3 describes a mobility model that captures direction-oriented movements. In Section 5.4, we design the SCR and elaborate on the two key features of SCR's predictive forwarding. Garbage collection and connectivity functions are also described, and the associated storage overhead is analyzed. Section 5.5 evaluates the SCR via simulation and numerical analysis, highlighting its performance improvements. Finally, we discuss the related work in Section 5.6 and conclude this chapter in Section 5.7.

5.2 System Description

This section describes the system that consists of a security context, SCRs, and multi-tier & overlay networks. When describing the system under study, we will use subscripts to identify objects; e.g. given $o_{s,i}$, i represents o's ID, and s represents the ID of a super-object to which o belongs.

5.2.1 Security Context

The contents of security context vary with the underlying protocol; e.g. the security context defined and used in MAP includes, authentication value pairs, mobile's ID, and other security information. The proactive keys defined in [58] can also be part of the security context, but the underlying protocol deals only with *intra-domain* handoffs. In most cases, a security context is derived from credential, and is distinct from it, since the security context is a variable part of credential, e.g. excluding a long-term key and detailed private information. Multiple replicas of a security context may be sent out to allow an MN to seamlessly hand off across all areas of access networks. Also included are other parameters for managing a security context, such as validity of lifetime for the context, validity of lifetime for each replica, and a removal-indicating bit for an obsolete replica (e.g. 1 denotes obsolete). Efficient management of the security context for *consistency* is required. Given below is a brief description of the SCR's roles in security-context management.

5.2.2 The Security Context Router

There are several roles for the SCR to play. First, it has to transfer a security context as an MN roams around: the MN's security context is transferred upon request (*reactive transfer*), or the replica(s) of its security context is(are) forwarded ahead of its arrival at a domain

based on the estimation of its future direction (*predictive forwarding*). Second, it must reduce the storage requirement by deleting obsolete replicas of security context (*garbage collection*). Third, each SCR has two interfaces: one for linking itself to other SCRs in a peer-to-peer manner, and the other for capturing its movement from the access nodes in a client/server manner. Multiple SCRs form an overlay network via their interfaces. Last, it should allow a *plug-in* authentication protocol, such as MAP and Kerberos [63], to run on a functional module whose architecture will be described in Section 5.4.

5.2.3 Multi-tier and Overlay Networks

For a wireless network built with heterogeneous technologies, one must make the best of the characteristics (e.g. coverage or cell and bandwidth) of each underlying technology. Such a network is an assembly of macro-, micro-, and pico-cell-based multi-tiers. A wired backbone and a number of base stations make up a cellular network. Each base station controls a cell, and a group of base stations are managed by a mobile switching center for circuit switching, or serving GPRS supporting node for packet switching, and an HLR/VLR for the authentication of MNs. When associated with a base station belonging to another mobile switching center, an MN registers with the VLR connected to the center. A WLAN is composed of access points and a back-end authentication server. A number of authentication clients under the control of each access point and a single home server make up an authentication system. When the MN is associated with the access point that another authentication server controls, its authentication request is dealt with in cooperation with the home server. A pico-net system is akin to a WLAN system — a number of slaves and an elected master connected to the back-end authentication server. SCRs are connected to VLRs/HLRs and the back-end authentication servers, and access nodes are regarded as

base stations, access points, and slaves.

Access networks are distinguished by different cell sizes as shown in Figure 5.1. Network *N* (e.g. a cellular network) covers a larger area than network *M* (e.g. a WLAN). Network *I* differs from network *J* in their operational policies. Networks *I* and *M* maintain themselves independently even if they are under the same administrator. Networks *L* and *O* are composed of tiny cells that form a private area network. Selecting which network to use depends on the availability of bandwidth along with a cost-effective policy. We use the term "domain" as an administration- and operation-independent area, which allows for clustering heterogeneous networks — shuffle and categorize them according to their application maintenance — and constructs a network with such clusters. Each network is basically regarded as a single domain, yet it can be divided into several disjoint (sub)domains — in Figure 5.1, *N* is divided into domains *P* and *Q*. Adjacent cells in the domain boundary do not need any lower-level agreement of association to understand protocols; i.e. a neighbor's ID, geographical topology/location, and protocol compatibility are unnecessary to be maintained. Instead, we introduce a *boundary neighbor register* that consists of a set of entities, allowing for the construction of a logical neighborhood. This will be detailed in Section 5.4.1.

An SCR represents a domain. A link between SCRs is introduced if needed; e.g. given scr_i in which an MN resided previously and scr_j to which the MN has just been bound, scr_i transfers its security context to scr_j upon request, and for the security-context transfer, there must be a secure connection between scr_i and scr_j. Such connections make up an overlay network, an undirected graph $G = (V, E)$, where $V=\{scr_1, \ldots, scr_n\}$ and E is a set of edges $e_{i,j}=(scr_i, scr_j)$. Domain I that scr_i controls is a set of access nodes, $\{an_{i,1}, \ldots, an_{i,n}\}$. The above overlay network is independent of geographical adjacency between access networks; e.g. in Figure 5.1, network K is adjacent to network L, while scr_k and scr_l

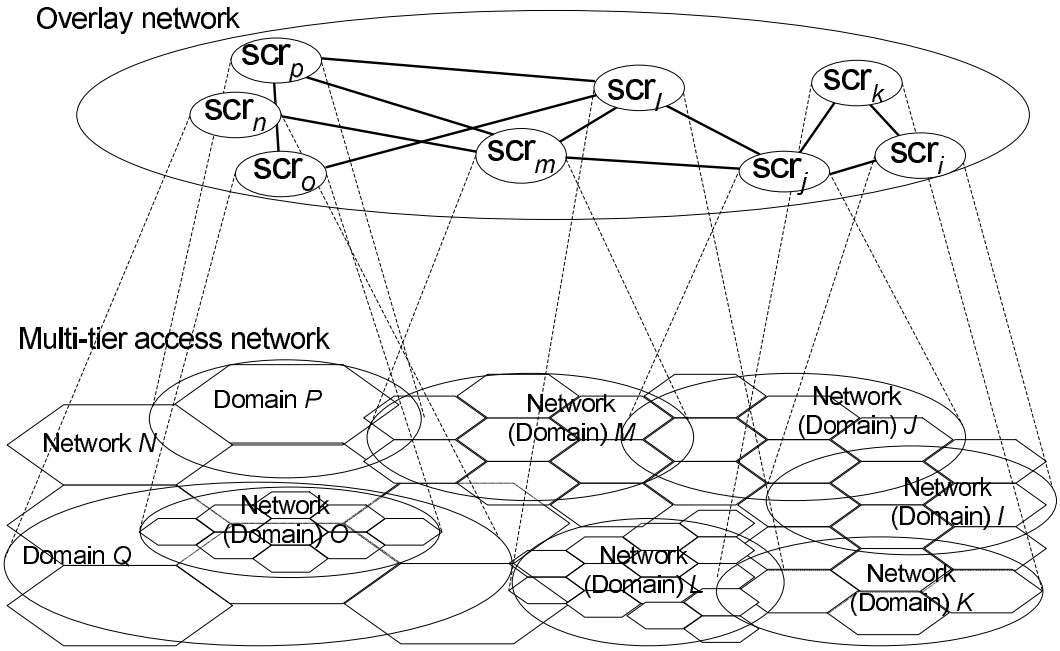


Figure 5.1: Multi-tier access and overlay networks. SCRs, each of which represents a feature domain, form an overlay network. A domain is covered by one or more SCRs.

are indirectly connected. In contrast, scr_l and scr_o are connected directly, but L and O are separated.

5.3 System Model

We model a network as a grid, and build a mobility model based on the observation that the direction of a mobile's movement is generally not ad hoc, as advocated by the global mobility model in [51].

5.3.1 Network Model

A group of access nodes are assumed to be distributed uniformly, and each group belongs to an SCR. As shown in Figure 5.2, an integrated network is represented as a grid of access

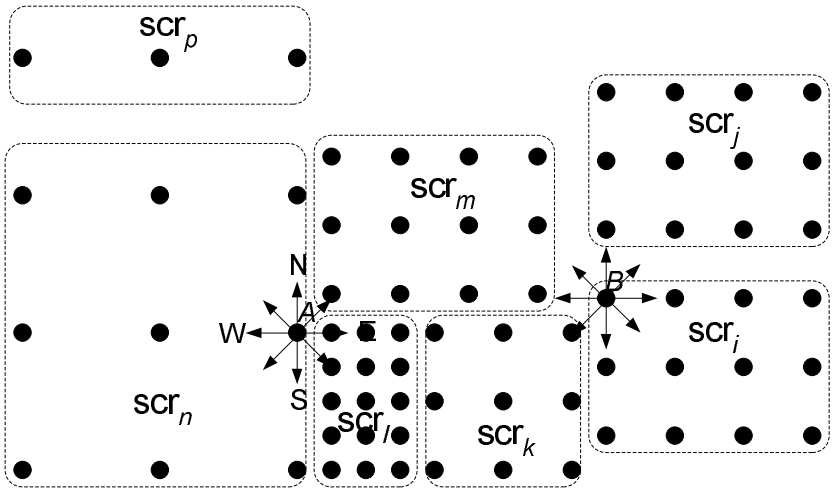


Figure 5.2: A grid of access networks; a point represents an access node, and MNs A and B may move in one of 8 directions.

nodes in which an MN can move and be associated with one of 8 surrounding access points. Given MN A bound to scr_n, observing the direction of its movement towards Southeast, it is most likely to move into the domain under scr_l's control, so that A's security context will likely be forwarded to scr_l. On the other hand, given MN B bound to scr_i, observing the direction of its movement towards North or Northwest, it is unlikely to move to the domain under scr_k's control. Therefore, scr_k will eventually be excluded from the set of receivers of B's replicas.

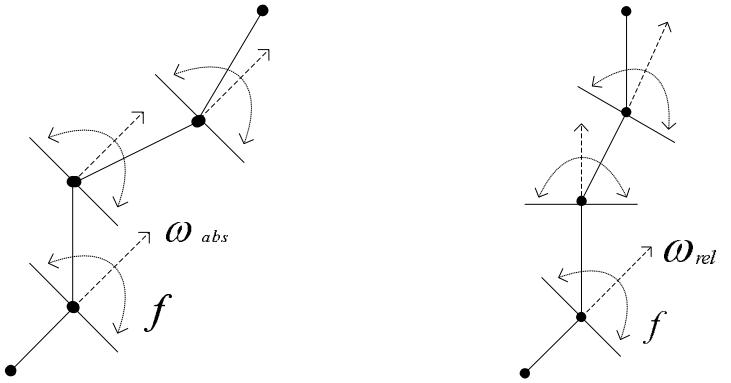
5.3.2 Mobility Model

Our mobility model is based on the fact that most MNs' movement directions are best characterized by TVAs, constructed by computing time-varying angles. MNs may move from their current location in any direction. The next movement direction is determined with current location l of $(x_j,\ y_j)$, a distribution f, and an anchor ω.

Two types of anchors are defined as a directional attitude.

- *Absolute anchor*: an MN's destination is known from the beginning, and the primary direction in which it intends to arrive does not change over time as shown in Figure 5.3-(a). The MN will eventually reach the destination.

- *Relative anchor*: the mobile's intended direction is calculated with current and previous positions; i.e. it is determined by the MN's past path as shown in Figure 5.3-(b). Given previous position $(x_i,\ y_i)$, we derive the relative anchor by computing $\tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right)$, while the absolute anchor is given initially.



(a) Absolute anchor: ω_{abs} (b) Relative anchor: ω_{rel}

Figure 5.3: Anchor-based determination of directions, where f and ω are the distribution and the anchor, respectively.

Provided that the directions of an MN's movement are observed and the probability of the MN taking a direction is relatively greater than the other directions, we generate a discrete distribution, f_D, with a random variable X of directions. Each direction is labeled with an index of the quantized angles

$$x^S = \{k : \forall\ \theta, \frac{2\pi}{n}(k-1) \leq \theta < \frac{2\pi}{n}k, k \leq n \in \mathbb{Z}\}. \tag{5.1}$$

By simplifying and categorizing the area, we compute f_D as follows. For the directions in

the angle of $(x_j,\ x_k]$, it is set to the probability

$$p = \sum_{(j,k]} f_D. \tag{5.2}$$

Within $(j,\ k]$, the MN's movement is pointed toward a most purposeful direction, which is set to be the conditional probability

$$q = f_D(j + \frac{k-j}{2})|p. \tag{5.3}$$

Subsequently, directions that are most unlikely to be taken are opposite to the intended directions; i.e. the angles of $(j',\ k']$ are represented as the interval $(j+\frac{n}{2}\ mod\ n, k+\frac{n}{2}\ mod\ n]$, which is set to be the conditional probability

$$r = \sum_{(j',\ k']} f_D| \sum_{\leq x} f_D - p. \tag{5.4}$$

The above derivation of the distribution is based on $n = 8$, but it can easily be generalized to the case of $n > 8$. As $p, q \to 1$, $r \to 0$, and $k - j \to 0$, the variance gets minimized, so that the MN's movement may appear unidirectional. As $p = 1, q, r = 0$, and $k - j \approx n/2$, f_D becomes uniformly-distributed over the interval $[-\pi/2, \pi/2]$, and in particular, at $p = 1/2$, so does it for all directions.

The MN's initial velocity is assumed to be a random variable with a normal Gaussian distribution truncated in the range of $(0,\ V_{max}$ km/h], and its variation draws from a uniform distribution within $\Delta_v\%$ of the mean velocity $E(V)$.

The TVA is represented as a matrix of $R \times A$, where R is the total number of records, and A the total number of quantized angles such that the MN hands off to one of access

nodes. Matrix **M** is written as

$$\mathbf{M} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,A} \\ \alpha_{2,1} & \ddots & & \vdots \\ \cdots & \cdots & \alpha_{r,\theta} & \cdots \\ \vdots & & \ddots & \vdots \\ \alpha_{R,1} & \alpha_{R,2} & \cdots & \alpha_{R,A} \end{pmatrix} \tag{5.5}$$

and $\alpha_{r,\theta}$ is given by

$$\alpha_{r,\theta} = \begin{cases} 1, & \text{if the MN has taken the direction} \\ 0, & \text{otherwise.} \end{cases}$$

The element $\alpha_{r,\theta}$ of the matrix indicates if the MN has taken the direction of a quantized angle.

Figure 5.4 shows the MNs' movement paths of A, B, C, D, E, and F. The anchor is initially set to Northeast, and the speed is set to be constant in order to highlight the effect of movement direction. Each time they are associated with an access node, the next position of the access node is determined by applying their specified distribution, anchor, and restricted angles available. A's path resembles mostly that of B which relies on a Gaussian function. C's path appears straight as driving on a straight highway. D's path results from prohibition of any backward turn as in [16, 82]. In contrast, E has randomness in all directions, where no regularity is observed. Significant portions of F's path are (close to) linear; i.e. no abrupt turns allow for extracting pattern sequences. We capture and classify such sequences to predict future directions. These examples will be used for performance evaluation.

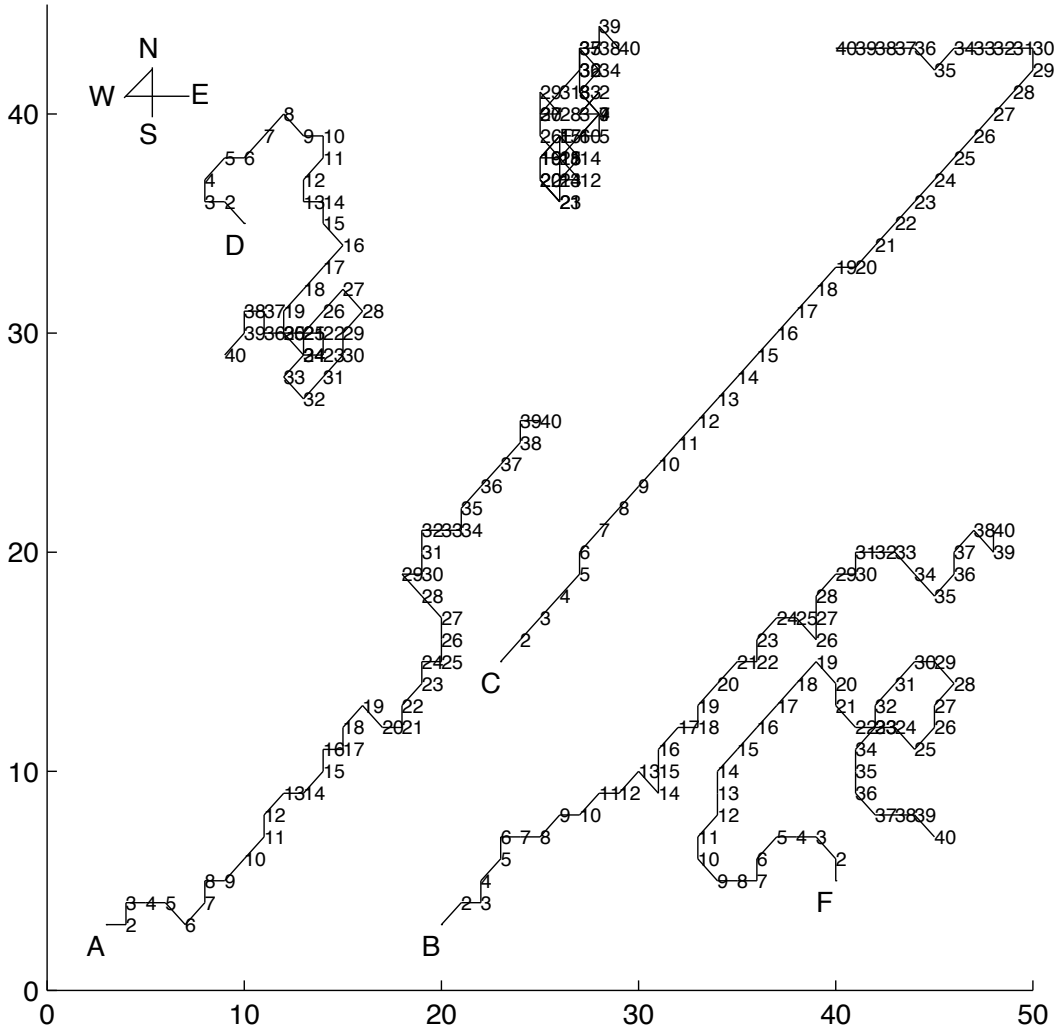


Figure 5.4: Comparison of a variety of distributions: A is the defined distribution $f_D(\theta \leq \pm\pi,\ \omega_{abs})$; B and C are Gaussian distributions $f_G(\theta \leq \pm\pi,\ \sigma = 1,\ \omega_{abs})$ and $f_G(\theta \leq \pm\pi,\ \sigma = 0.5,\ \omega_{abs})$, respectively; D and E are uniform distributions $f_U(\theta \leq \pm\pi/2,\ \omega_{rel})$ and $f_U(\theta \leq \pm\pi,\ \omega_{rel})$, respectively; F is $f_G(\theta \leq \pm\pi,\ \sigma = 1,\ \omega_{rel})$, where ω_{abs} and $\omega = rel$ denote absolute and relative anchors, respectively.

5.3.3 Edit distance

To see (dis)similarities of pattern sequences, one sequence is edited to become the same as the other; i.e. the *edit distance* to determine the resemblance between two sequences [32] is computed as follows. Given an observed sequence $a_1 \cdots a_m$, and a pattern $b_1 \cdots b_n$, we construct an $(m+1) \times (n+1)$ matrix $(d(a_1 \cdots a_n, b_1 \cdots b_n))$, with (i, j) ranging from $(0, 0)$ to (m, n). If $i = 0$ or $j = 0$ meaning that $a_1 \cdots a_i$ or $b_1 \cdots b_j$ is an empty sequence, then the matrix values are $d(1,\ 1) = 0$, $d(1,\ b_1 \cdots b_i) = 0$, and

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

To compute $d(a_1 \cdots a_i,\ b_1 \cdots b_j)$, consider each path from $(0, 0)$ to (i, j) as follows:

$$d(a_i,\ b_j) = \begin{cases} d(a_{i-1},\ b_{j-1}) & \text{if } a_i = b_j \\ min \begin{cases} d(a_{i-1},\ b_{j-1}) + C_C, \\ d(a_{i-1},\ b_j) + C_D, \\ d(a_i,\ b_{j-1}) + C_I \end{cases} & \text{if } a_i \neq b_j. \end{cases} \tag{5.6}$$

All costs are set to 1 in generic methods [32], but we integrate the "degree of matching" into the cost, i.e. the cost increases as the matching proceeds further. Our proposed costs C_C, C_D, and C_I will be defined in the next section.

5.3.4 Pattern Classification

A *training set* of data is built where we observe the outcome and feature measurements for a set of direction patterns. This data will be used to construct a prediction model for the outcome of new unseen TVAs. There are two simple but powerful prediction meth-

ods: the linear model fit by least squares and the k-nearest neighbor prediction rule [35]. The linear model is known to yield stable but possibly inaccurate predictions, while the method of k-nearest neighbors makes prediction accurate but can be *unstable* — any particular subregion of a *decision boundary* depends on a handful of input points and their particular positions. These two are complementary, and will be used to classify patterns for prediction.

In the least-squares methods, given a column vector of inputs X=(X_1, …, X_p), we predict the output Y via the linear model in vector form as an inner product

$$\hat{Y} = X^T\hat{\beta}, \tag{5.7}$$

where X^T denotes the transpose of vector X, and $\hat{\beta}$ denotes the prediction of bias. We fit the linear model to a set of training set by using the least-squares method. We choose coefficients β by minimizing the residual sum of squares

$$\begin{aligned} &RSS(\beta) = (y - X\beta)^T(y - X\beta), \\ &X^T(y - X\beta) = 0, \end{aligned} \tag{5.8}$$

where y is the output in the training set. If X^TX is non-singular, then the unique solution is given by

$$\hat{\beta} = (X^TX)^{-1}X^Ty, \tag{5.9}$$

and the fitted value at the i-th input x_i is $\hat{y} = x_i^T\hat{\beta}$, which is the prediction. On the other hand, the k-nearest neighbor fit for $\hat{Y}$ is defined as:

$$Y(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i, \tag{5.10}$$

where $N_k(x)$ is the neighborhood of x defined by the k closest points x_i in the training set. In other words, we find k observations with x_i closest to x in input space, and compute the mean of their responses.

To classify the outputs of *qualitative* data of TVAs, we transform them into numerical labels, which allows for quantitative distinction, by computing

$$\begin{aligned} X_1 &= \log \textstyle\sum_{i=1}^{n} I_\theta(\alpha_{n-i+1,\theta} \neq 0) 2^{n-i}, \\ X_2 &= \log \tfrac{1}{n-1} \textstyle\sum (I_\theta - E(I_\theta))^2, \end{aligned} \tag{5.11}$$

where $I_\theta(x)$ denotes an index of θ satisfying x. In the representation of X_1, the multiplication of 2 to n power reflects the depth of the vector, which will be detailed when we discuss the cost of edit distance; the edit distances between TVAs are slightly different from those between the transformed ones.

5.4 Design of SCR

This section elaborates on the design of an SCR and its functional modules. Based on Figure 5.5 that illustrates the underlying architecture of SCR, we first outline data sets correlated with their access functions, and then delineate the functional modules each of which is self-reliant. Finally, we quantify the storage overhead of replication, and analyze optimization of the overhead.

5.4.1 Data Sets with Access Functions

There are four data sets maintained and accessed via the corresponding access functions according to the intended functionality.

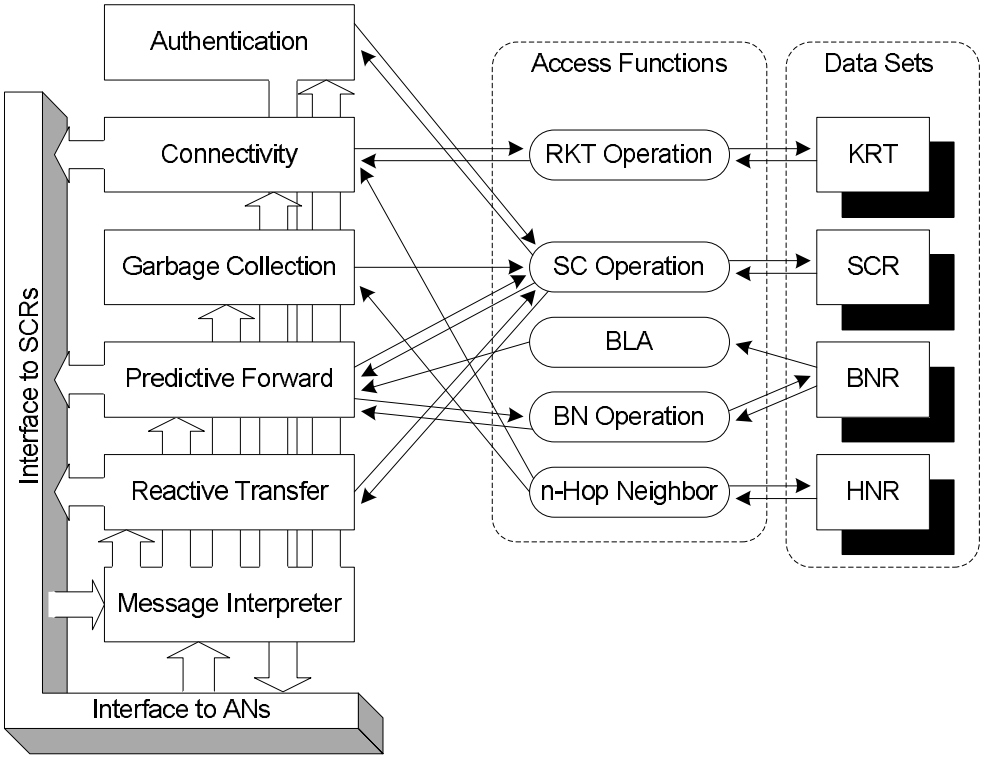


Figure 5.5: SCR architecture. It consists mainly of four data sets, the corresponding access functions, and six functional modules coordinating with interfaces to SCRs and ANs.

- R_i, key-associated route table (KRT), is a list of entities each of which consists of the fields of SCR's ID, the corresponding accessible address, and cryptographic key; e.g. scr_i looks up the table to find scr_j, and encrypts or decrypts exchanged messages with the corresponding key which may be a symmetric or public key depending on the type of secure mechanism to be used. The description is detailed in Chapter 4.

- S_i, security context register (SCR), is a set of security contexts. The security context operations include functions to read, add, delete, and update the security context from/to S_i. In addition to the security context, its replica is also stored temporarily, and if the replica is used before its validity is expired, then the validity is updated with that of the security context, and the removal-indicating bit is set to 0. Otherwise, it is set to 1, which signifies the removal of the replica to the garbage-collection module.

- B_i, boundary neighbor register (BNR), is a set of triples, $\{(an_{i,1}, scr_j, an_{j,l}), (an_{i,2}, scr_j, an_{j,m}), \ldots, (an_{i,k}, scr_j, an_{j,n})\}$, each triple of which results from an MN's cross-domain handoff, e.g. from $an_{i,k}$ to $an_{j,n}$ that belongs to scr_j. It provides logical conjunctions of access nodes in both sides of the boundary. In addition, the access nodes in the boundary can be extracted by collecting the first value of each triple in B_i, i.e. L_i=$\{an_{i,1}, an_{i,2}, \ldots, an_{i,k}\}$. Therefore, given an MN associated with $an_{i,k} \in L_i$, scr_i presumably observes the likelihood of crossing the domain boundary.

- H_i, n-hop neighbor register (HNR), is a set of IDs of SCRs that are linked within the n-hop boundary from scr_i. So, H_i is used to find neighbors in scr_i's vicinity and to propagate a *removal* message to neighbors within the n-hop boundary. It is constructed as follows:

function h(i, n)

if $n \leq 1$ **then**

Return $\{x\colon (i, x) \in E\}$

else

Return $\{x\colon \forall\, k \in \mathrm{h}(i, n-1), (k, x) \in E\}$

end if

5.4.2 Functional Modules

Message Interpretation and Scheduling: There are two interfaces to ANs and SCRs, represented as buffers ϵ^S and ρ^S, respectively. Provided that MN A has just been associated with $an_{j,n}$ in scr_j from $an_{i,k}$ in scr_i, messages $\epsilon \in \epsilon^S$ and $\rho \in \rho^S$ essentially contain the fields of IDs A, j, n, i, and k. In some cases, the information on previous associations (e.g. i and k) may be unavailable, caused by switch-off-and-on, or a loss of connection to the MN. Message ϵ from $an_{j,n}$ is interpreted as one of the following three cases.

C1. The MN has just been associated with $an_{j,n}$, coming from $an_{i,k}$ in scr_i, and its security context transferred from scr_i either reactively or proactively is ready for authentication; its security context is found in S_j. Either the MN is authenticated or mutual authentication takes place, depending on the security protocol used. After a successful authentication of the MN, scr_j informs scr_i of such binding, by sending `Successful Authentication` message, of type 2. Due to the fact that the MN associated with $an_{j,n} \in L_j$ can return, the predictive forwarding follows. So, scr_j sends `Predictive Forwarding` message, of type 1.

C2. The MN has just been associated with $an_{j,n}$, but there exists no corresponding security context. So, scr_j requests its security context from scr_i; it performs a

reactive transfer, sending `Reactive Transfer` message, of type 3.

C3. The MN has handed off from $an_{j,n}$ to $an_{j,x} \in L_j$, which means to scr_j that the MN is likely to cross the boundary, so that it performs the predictive forwarding by sending `Predictive Forwarding` message, of type 1.

Message ρ from neighboring SCRs is interpreted as one of the following five types.

T1. As a reaction to propagation of the replica corresponding to Cases 1 and 3 of `Predictive Forwarding` message, scr_x adds it to S_x; if there exists an obsolete replica, the most recent one is updated.

T2. As a response to an acknowledgement corresponding to Case 1 of the `Successful Authentication` message, scr_i deletes the corresponding security context from S_i. It may be scr_i that sends out the *removal* message, of type 5, for propagated replicas to its neighbors, $h(scr_i, n) \subset H_i - \{scr_j\}$. If scr_j sends the removal message instead, it signals the message to H_j. Note that if the MN resided in scr_i, this case would not occur, which results in the state of the replicas of its security context in scr_j being *obsolete*. We rely on garbage collection to remove such obsolete replicas.

T3. As a response to the request for the security context corresponding to Case 2 of the `Reactive Transfer` message, scr_i sends the security context to scr_j with message type 4, and then deletes it from S_i. Meanwhile, it adds this association $(an_{i,k}, scr_j, an_{j,n})$ to B_i.

T4. As a response to the message of type 3, scr_j receives and adds the corresponding security context to S_j while adding the association $(an_{j,n}, scr_i, an_{i,k})$ to B_j.

T5. The garbage-collection module is executed in case of the removal message propagated.

Reactive Transfer: To fetch the security context, scr_j extracts information on scr_i from ϵ. If no information on the previous association is found, it performs an n-hop neighbor search. If scr_i is requested and finds the security context in S_i, it responds to scr_j. Both scr_i and scr_j then add this association to their BNRs. Meanwhile, a 1−hop connection is established if a cryptographic key is exchanged successfully, by updating their KRT.

Predictive Forwarding: scr_i propagates as many replicas of security context as the cardinality of the following set:

$$G(x) = \{e_2 : \forall\ e = (e_1, e_2, e_3) \in B_i,\ x \equiv e_1\}. \tag{5.12}$$

Learning that the MN from $an_{i,k}$ is highly unlikely to be bound to $\{scr_k, \ldots, scr_l\} \subset G(m)$, scr_i will exclude the SCRs in such a set, i.e. we minimize the set of SCRs to which the replicas of security context will be forwarded. For this optimization, we utilize approximate pattern matching and statistical estimation.

- *Cost of Edit Distance*: We here define the cost of edit distance in Eq. (5.6) of Section 5.3.3. C_C is the cost of changing b_j in $b_1 \cdots b_n$ by a_i in $a_1 \cdots a_n$, and is computed as:

$$C_C = \begin{cases} 1 + \alpha & \text{if } a_i \leq b_j \pm 1, \\ 2 + \alpha & \text{if } a_i \leq b_j \pm 2, \\ \infty & \text{otherwise,} \end{cases}$$

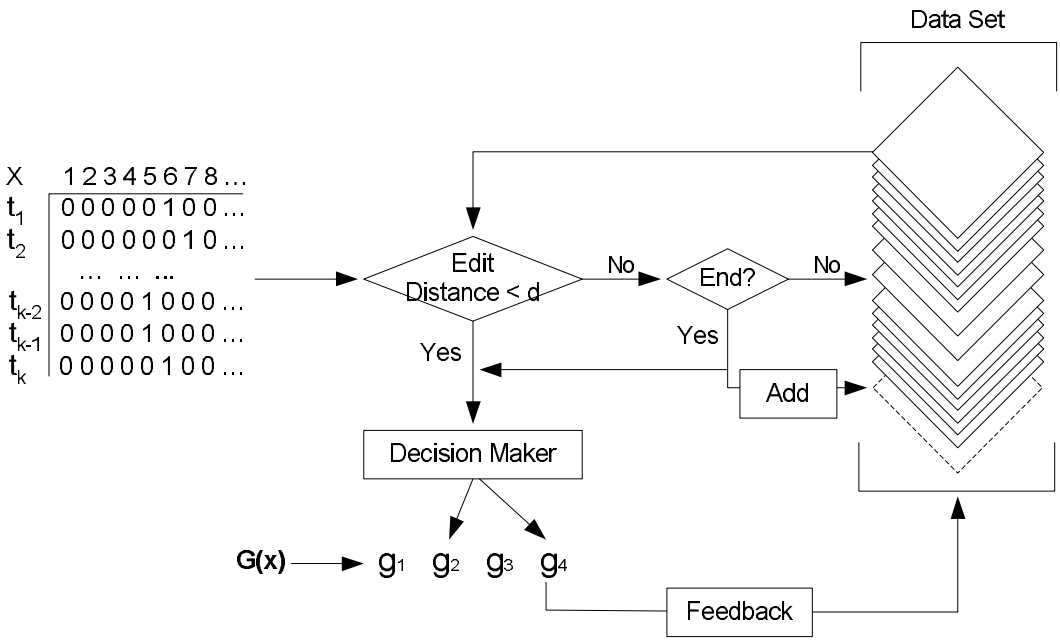


Figure 5.6: Estimate of replica forwarding with approximate pattern matching and statistical estimation.

where $\alpha = \frac{\overline{j}}{n}$ is the rounded-off weight, and increases as $j \to n$. C_D is the cost of deleting b_j. If $b_1 \cdots b_i$ is empty, there is no cost.

$$C_D = \begin{cases} 0 & \text{empty,} \\ 1 + \alpha & \text{otherwise.} \end{cases}$$

C_I is the cost of inserting a_i at position i, computed as:

$$C_I = \begin{cases} 1 + \alpha & \text{if } a_i \leq b_j \pm 1, \\ 2 + \alpha & \text{if } a_i \leq b_j \pm 2, \\ \infty & \text{otherwise.} \end{cases}$$

Our gradually-weight-imposed edit distance is different from Liu's [51] in that the weight cost varies with sensitivity to deviations from the beginning to the end of the

Table 5.1: Comparison of a weight-imposed method with Liu's and generic methods

Edit distance	Our method	Liu's	Generic method [32]
d_1(12367, 12345)	4	∞	2
d_2(62467, 34567)	3	3	3
d_3(34684, 34567)	6	∞	3
d_4(34345, 12345)	3	∞	2
d_5(32345, 23234)	1	1	3

sequence of a pattern; our method yields more robust weight cost in the sense that the MN's movement changes near the boundary affect a future position more than those away from it. In Table 5.1, comparing d_1 with d_2 in our method, deviations at the end of the sequence impose a heavier weight on cost. Comparing d_1 with d_3, our method allows for differentiation of deviations at the end whereas that in [51] does not. In case of d_5, our method recognizes similarity better than the generic method.

- *Statistical Estimation*: As illustrated in Figure 5.6, the estimation procedure begins with recording the TVA of an MN in the matrix form of Eq. (5.5). Using the Chebyshev inequality theorem [68], we define the distribution f_S of a random variable X in $G(x)$ with a finite mean μ and a finite variance σ^2. For any value of dropout rate $\delta_p > 0$, we obtain

$$P(|X - \mu| \geq \frac{\sigma}{\sqrt{\delta_p}}) \leq \delta_p. \tag{5.13}$$

We set $\sigma = 1$ for any f_S, showing that irrespective of the type of f_S, the equation works effectively. It allows the decision-maker to exclude the values of x that are outside $(\mu - 1/\sqrt{\delta_p}, \mu + 1/\sqrt{\delta_p})$ when the replicas of security context are to be forwarded. If the MN is bound to $x \in X$ eventually, $G(x)$ is updated. However, the feedback may change the type of f_S; we will show numerically the impact of such

variations in the evaluation section.

Garbage Collection: A number of replicas of security context will be piled up in S_x, and the amount of required storage depends on the number of MNs. One may argue that there is a large enough storage to keep all obsolete replicas. However, since there could be many MNs joining and leaving the service, we take a *soft-state* approach to lowering the storage overhead; for this there are a variety of approaches augmented with functional parameters [39], but we rely on removal of obsolete replicas such that scr_x scans S_x, selects the replicas with the removal-indicating bit set to 1, and removes the expired ones of them. A timestamp t_s is used in forwarding the replicas to check if they are expired. Given the current time t_c, the elapsed time $t_c - t_s$ is checked periodically, and if it is greater than a threshold value δ_t, then the corresponding replica is removed from S_x. The value of δ_t is set large enough to avoid a false positive cache hit; e.g. $C \cdot E(SojournTime)$, where C is a constant and E is the mean of MNs' sojourn times in a cell.

Connectivity: Given scr_i and scr_j that are at first *logically* disconnected from each other, when scr_i to which a randomly chosen MN is initially bound finds no neighbor, it contacts the MN's home server to obtain its security context. When the MN moves and is bound to scr_j, scr_j searches for n-hop neighbors, and if found, scr_i responds with its security context. Meanwhile, either scr_i or scr_j informs the home server of such rebinding of the security context. Otherwise, scr_j contacts the home server directly to obtain (the location of) the security context. During the rebinding, scr_i and scr_j make a logical link between them.

Authentication: The SCR allows a plug-in authentication protocol — which is required to create a security context — to run on the authentication module. On behalf of the home server, the SCR can execute the mobile authentication/authorization procedure based on the created security context. The execution policy (one-way/mutual authentication, or resource grant) is contingent upon the specification of the protocol.

5.4.3 Reduction in Storage Overhead

When n MNs are associated with access nodes of scr_x in the boundary, n replicas of security context are forwarded to $|G(x)|$ neighboring SCRs. Assuming that the probability p that an MN will cross the boundary is equal to that of the others, the storage overhead of n replicas is

$$S = |G(x)|n - p \times n. \tag{5.14}$$

It varies with the elapsed time, e.g. initially at time t_0, $S_{t_0} = |G(x)|n_{t_0}$, $S_{t_1} = |G(x)|n_{t_1} + S_{t_0}$ in the first unit of time, $S_{t_2} = |G(x)|n_{t_2} + S_{t_1}$ in the second unit of time, and so on. In n units of time, we have

$$S_{t_n} = |G(x)|n_{t_n} + (|G(x)| - p) \sum_{i=1}^{n} n_{t_{i-1}}, \tag{5.15}$$

where n_{t_i} is the number of replicas in the i-th unit of time, t_i. Due to the fact that there may already be obsolete replicas in S_x (and if so, it is just updated), the amount of occupied storage $\sum n_t$ appears at first a linear- but down-curve-like increase over time. Therefore, $\sum n_t$ is viewed as a multiplied cumulative function represented as $N \times F(X)$, where N is the total number of MNs and $F(X)$ is a Gamma cumulative distribution with a random variable

X of $t^{\dagger}$, in the reverse order of t. Then, we obtain

$$\begin{aligned} S_{t_n} &= |G(x)|N \times f(t_0^{\dagger}) + (|G(x)| - p)N(F(t_{n-1}^{\dagger}) - F(t_0^{\dagger})) \\ &\approx (|G(x)| - p)N \times F(t_n^{\dagger}). \end{aligned} \tag{5.16}$$

To reduce the storage overhead, we clean up S_x according to δ_t. The threshold should be greater than a timestamp in any replica; e.g. $C \cdot E(SojournTime)$, as discussed earlier. Since scr_x applies δ_t to all, it should be large enough to avoid a false positive cache hit. Therefore, we obtain

$$S_{t_n} = (|G(x)| - p)N \times F(\delta_t). \tag{5.17}$$

On the other hand, in statistical estimation, the decision-maker forwards the replicas to m neighboring SCRs belonging to $G(x)$ that are inside the interval $(\mu - 1/\sqrt{\delta_p}, \mu + 1/\sqrt{\delta_p})$. Let q be the conditional probability that some of MNs are bound to some of SCRs in $G(x)$ assuming that all MNs are bound to the SCRs in $G(x)$. It is thus $q = \sum_{X \in m} P(X)|p$. Then, we obtain

$$S_{t_n} = (m - p \times q)N \times F(\delta_t). \tag{5.18}$$

Note that there is a tradeoff between the storage overhead and computational complexity.

5.5 Evaluation

This section evaluates the key features of predictive forwarding. First, we describe our evaluation methodology, and then analyze the evaluation results. We also numerically analyze the reduction in storage overhead.

5.5.1 Evaluation Methodology

Our performance evaluation is done on a platform that consists of a software implementation of SCRs forming an overlay network, and an emulation of MNs as well as a multi-tier access network. The multi-tier network is realized as a 25×25 grid, each point of which represents an access node. The grid is equally partitioned into 25 domains, each of which is under an SCR's control. 150 MNs are distributed uniformly, and their handoffs among access nodes are emulated according to the mobility model described in Section 5.3.2; e.g. the association of an MN with an access node invokes a *trigger* message to the corresponding SCR via a TCP port (a reliable link), and an SCR communicates with another via a UDP port (a best-effort link). However, for simplicity, we ignore the impact of packet losses on performance evaluation. In order to synchronize the events taking place between the overlay and access networks, we define *tick*, a unit of logical time. A handoff takes t ticks, the operation of each module in the SCR consumes 1 tick, and the message exchange between SCRs takes less than t ticks.

Table 5.2: Parameters used in evaluation

Defined parameters	Values
length of TVA	5
δ_p in Eq. (5.13)	0.5
p, q, r, s in f_D	0.9, 0.45, 0.01, 0.99
n in Eq. (5.1)	8
δ_d	3

Table 5.2 shows the values of parameters used throughout the evaluation. TVAs are built by recording the variations of angles for $5 \cdot t$ ticks. The direction θ is quantized into 8 regions. Unless specified otherwise, we will use these values.

5.5.2 Evaluation Results

Effectiveness of Proactive Caching Each handoff invokes a reference to the security context. Figure 5.7 shows the effectiveness of advance forwarding of security context as the completion ratio of BNR increases. This result comes from flooding of security contexts — 'turning up' the thresholds in pattern matching and statistical estimation into maximum, so as to highlight the correlation of the BNR completion with possible handoff disruptions. An MN initially bound to scr_{19} enters the domain of scr_{13} after making a local handoff. Due to the unavailability of security context, a reactive transfer, which takes 3 units of time, is activated. However, other MNs contribute to BNR completion gradually over time, and therefore, the MN's move from scr_6 to scr_7 and to scr_{13} via scr_8 takes 2 units of time, which is the same as the latency of a local handoff. Even an incomplete BNR makes an impact on proactive caching (for which 40% of BNR completion allowed in the figure). The more MNs, the quicker the BNR is completed.

Pattern Recognition The performance of approximate pattern matching varies with a variety of distributions. Figure 5.8 shows configurations of similarity that result from matching the captured TVAs with patterns at an edit distance. If matched with any of classified patterns (circled points), a captured TVA appears as a dotted point; otherwise, it is added as a new pattern. The density of a group of dotted points and the number of such groups indicate the degree to which patterns are recognized, and TVAs similar to others as various patterns are generated. The absolute anchor-based direction patterns in the left column of the figures assume greater similarity than the relative anchor-based ones in the right column of the figures except for the results from uniform distributions. The f_D-based direction patterns in Figures 5.8-(a) and (b) are classified well, compared to the f_G-based ones, of $\sigma = 1$, in Figures 5.8-(c) and (d). As expected, the f_G-based directions,

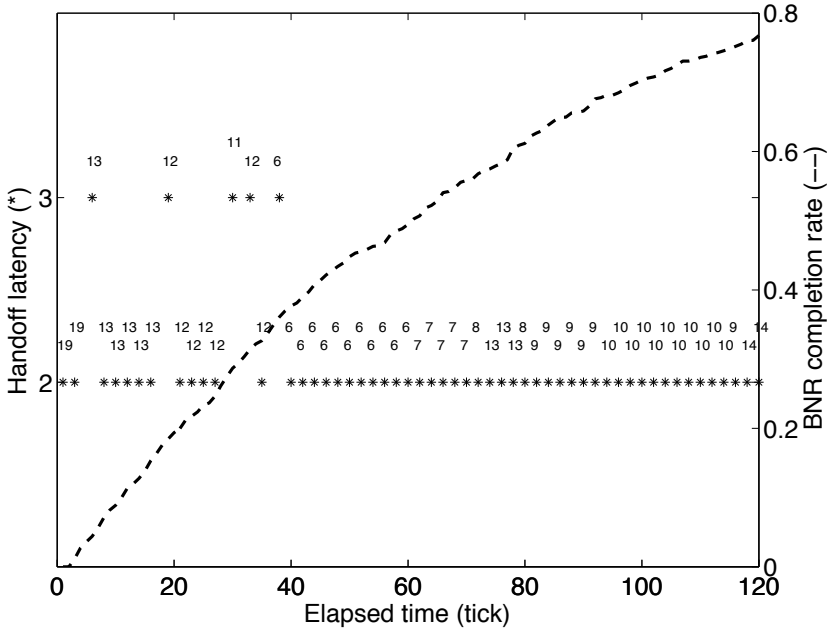
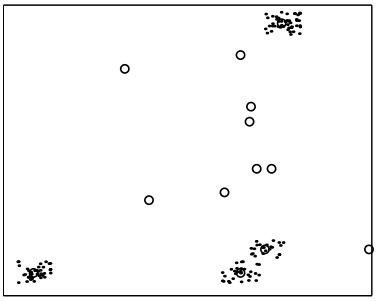


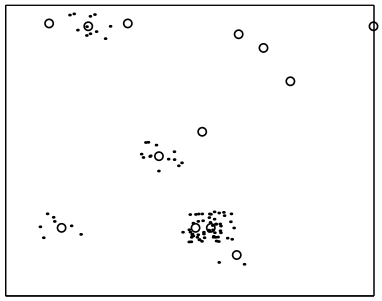
Figure 5.7: The effectiveness of advance forwarding. Numbers over asterisks indicate IDs of SCRs to which an MN is bound and handing off. The dash line is a best-fit to points that represent a completion rate at each tick, which are omitted for clarity.

of $\sigma = 0.5$, in Figures 5.8-(e) and (f) show strong homogeneity, while the uniform-based ones in Figures 5.8-(g) and (h) lack homogeneity, especially Figure 5.8-(h) corresponding to the pure ad hoc case.

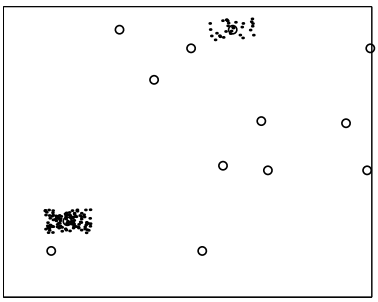
Pattern Prediction We observe n MNs, of classified pattern, that have initially been associated with an_5 in scr_{13} crossing the domain boundaries and being re-associated with access nodes in scr_{14} and scr_8. We classify their associations into two classes: the one bound to scr_{14} and the other to scr_8, each of which has two groups (associated with an_1 and an_6 or an_{24} and an_{25}) — these groups can be classified recursively. We build a training data of these patterns that can be seen in Figure 5.9-(a). The two classes are represented numerically by code, of 0 or 1, and then fit by linear regression and by 3-nearest neighbor averaging as shown in Figures 5.9-(b) and (c), respectively. The predicted classes are separated by the decision boundary.



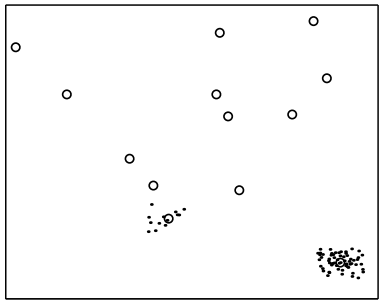
(a) $f_D(\theta \leq \pm\pi,\ \omega_{abs})$



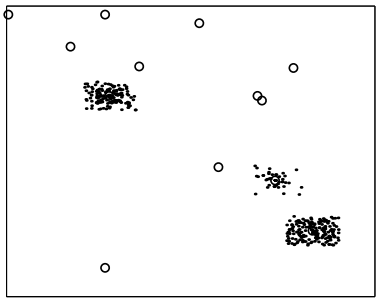
(b) $f_D(\theta \leq \pm\pi,\ \omega_{rel})$



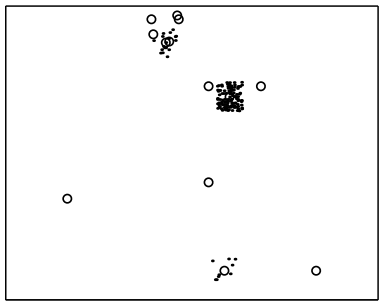
(c) $f_G(\sigma = 1,\ \theta \leq \pm\pi,\ \omega_{abs})$



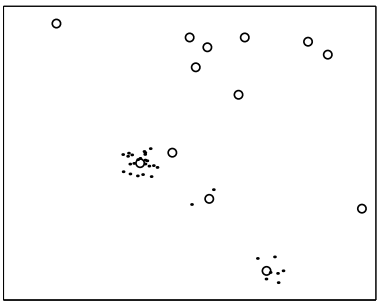
(d) $f_G(\sigma = 1,\ \theta \leq \pm\pi,\ \omega_{rel})$



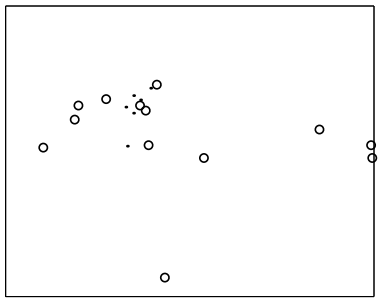
(e) $f_G(\sigma = 0.5,\ \theta \leq \pm\pi,\ \omega_{abs})$



(f) $f_G(\sigma = 0.5,\ \theta \leq \pm\pi,\ \omega_{rel})$



(g) $f_U(\theta \leq \pm\pi/2)$



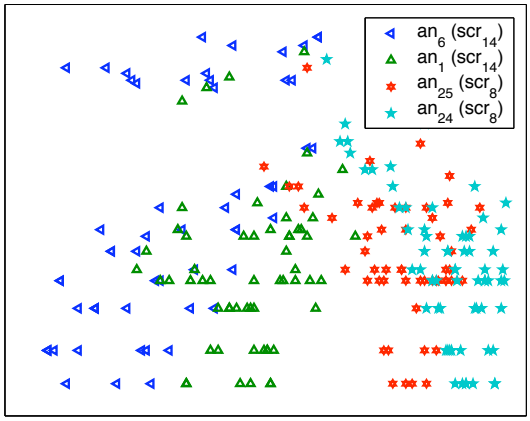
(h) $f_U(\theta \leq \pm\pi)$

Figure 5.8: Comparison of pattern recognitions with various distributions. A circled point is represented as a classified pattern, and dotted points around the circled are the ones matched with those at an edit distance. All points are displayed via Eq. (5.11); the range of a group of dotted points corresponds to the edit distance. f_D, f_G, and f_U are discrete, Gaussian, and unform distributions, respectively.

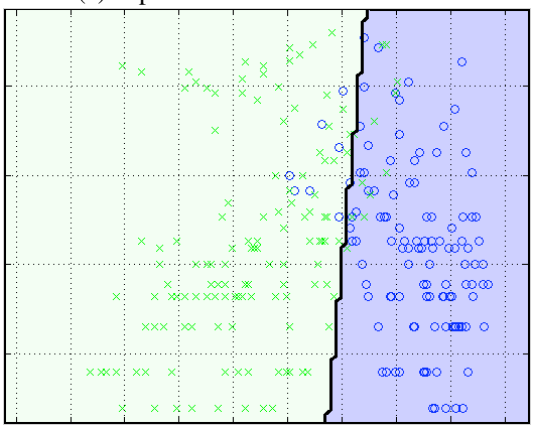
The regions overlap inevitably, signifying that even two similar patterns may take different directions. There are several misclassifications on both sides of the decision boundary. Note that these can be errors on the training data itself — the distance among training data is not exactly equal to that among classified patterns. The linear decision boundary from the least squares fit is stable, while that from the 3-nearest neighbor procedures is irregular, but might be optimal. However, classification errors of the former and the latter are almost the same (equal to 0.083 and 0.087, respectively). Consequently, the two classes are distinguished clearly — each of them has individual mean and variance distribution — and prediction by pattern classification is appropriate for our approach to approximate pattern matching. Therefore, provided that a captured TVA of an MN is found matched with any pattern in the training data, we predict the future direction of the MN via the decision boundary.

5.5.3 Numerical Analysis

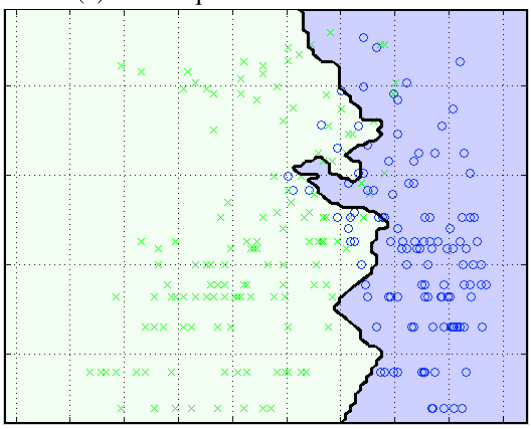
Impact of soft states on storage overhead The efficiency of the soft-state approach varies with types of the cumulative distribution of replicas, $F(\delta_t)$, in Eq. (5.17). Despite the optimal δ_t, there is a fundamental limit in reducing the storage overhead, because δ_t should be adjusted to be greater than any validity lifetime of the received replicas to avoid false-positive cache hits. We assess the impact of F on the storage overhead augmented with the parameter from $\beta = 2$ to $\beta = 8$, introducing a smooth increase to upcoming new replicas over time. In Figure 5.10, with optimal δ_t, the smooth increase reduces the storage overhead by more than 90%. However, in case of an abrupt increase ($\beta = 2$), only an about 60% reduction is achieved, which is, in turn, difficult to tune δ_t to be optimal in the sense that the achievement of most of significant reductions results from more than 90% of



(a) A pattern classification



(c) Least-squares fit



(b) 3-nearest neighbor fit

Figure 5.9: Pattern classification and predictions

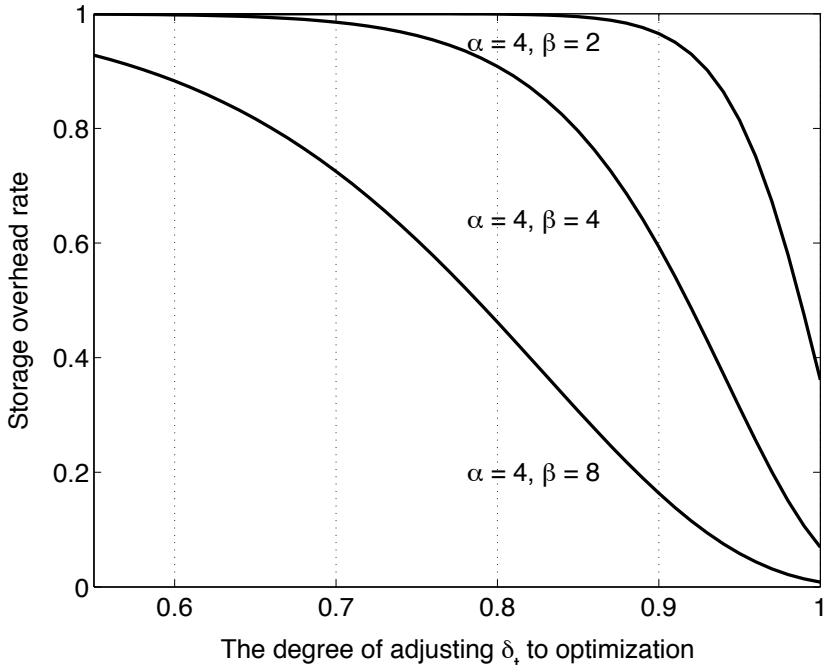


Figure 5.10: The storage overhead while adjusting δ_t to be optimal. Given fixed α in a cumulative Gamma distribution $F(X)$, the larger the value of β, the lower and wider the convex shape of the distribution along with the x-axis.

optimally-tuned δ_t.

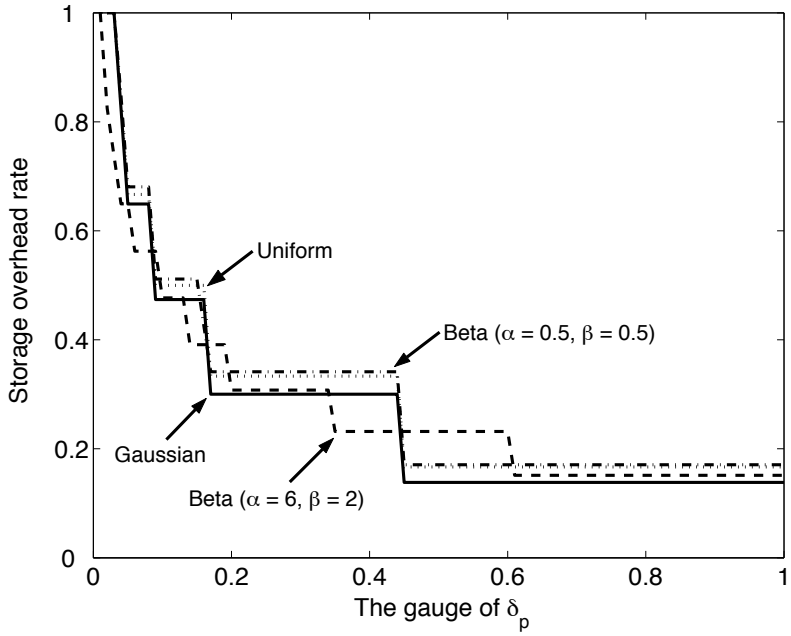
Effectiveness of statistical estimation Statistical estimation helps decrease the storage overhead; the narrower the range of target SCRs to which replicas are forwarded, the lower the storage overhead, minimizing the false negative rate — the rate of misjudging the targets. We evaluate the statistical estimation method with various types of f_S: normal Gaussian and Beta (α=6, β=2) distributions assume a highly-integrated density in choosing some SCRs in $G(x)$, while uniform and Beta (α=0.5, β=0.5) distributions assume the equivalent and divided chances, respectively. Figures 5.11-(a) and (b) show the correlation between the storage overhead and δ_p, affecting Eq. (5.18), and the corresponding false negative rate, respectively. Note that the number of selected SCRs falling in any computed confidence interval is always an integer. The estimation method is shown to operate

as expected, achieving a significant reduction in the storage overhead under any distribution. With $\delta_p \leq 0.2$, Beta (α=0.5, β=0.5) and uniform distributions also fit the estimation method (in Figure 5.11-(a)), but their false negative rate increases steeply (in Figure 5.11-(b)). Nevertheless, the overall risk under all distributions is kept lower than 5% of the rate. In general, the effectiveness of the estimation method is maximized when f_S assumes a normal Gaussian distribution.

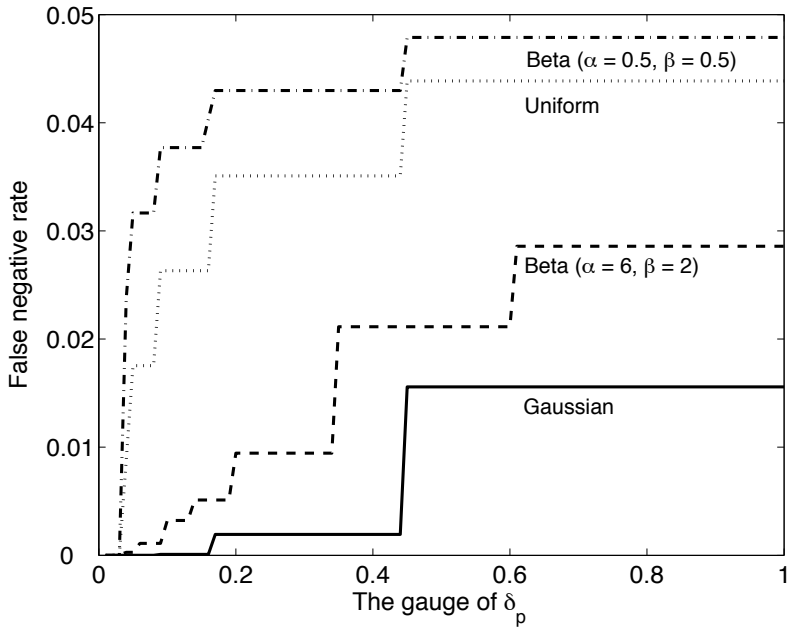
Four is the minimum $|G(x)|$ for effective statistical estimation The performance/risk varies with the number of neighbor SCRs, $|G(x)|$. Figure 5.12 shows the storage-savings gain when f_S assumes a normal Gaussian distribution, and the corresponding false negative rate as $|G(x)|$ increases. When $|G(x)| \geq 4$, the statistical estimation method operates effectively, achieving the minimum gain of 40%; the lower bound of effectiveness for a normal Gaussian distribution is 4. The false negative rate decreases as $|G(x)|$ grows. We also verified the effectiveness for uniform and two Beta distributions (α=6, β=2, and α=0.5, β=0.5). Uniform and Beta (α=0.5, β=0.5) distributions yielded results similar to those of a normal Gaussian distribution, while for a Beta (α=0.5, β=0.5) distribution, it operates effectively when $|G(x)| \geq 3$.

5.6 Related Work

Loughney *et al.* [52] defined a context-transfer protocol in order for mobiles to operate with minimal disruption. This context is broad, and differs slightly from ours in that our security context encompasses a consistency constraint. The majority of its functionality is concerned with reactive transfer. Mishra *et al.* [57] presented proactive caching of context among access points within a wireless network. They use a neighbor graph with a set of vertices representing access points joined by edges representing mobiles' paths. They



(a) Decrease of storage overhead



(b) The rate of false negative decisions

Figure 5.11: The results of statistical estimation from four plausible distributions; a Beta distribution with α=6 and β=2 has a right-aligned concave down shape and a Beta distribution with α=0.5 and β=0.5 is concave up with a unique minimum.

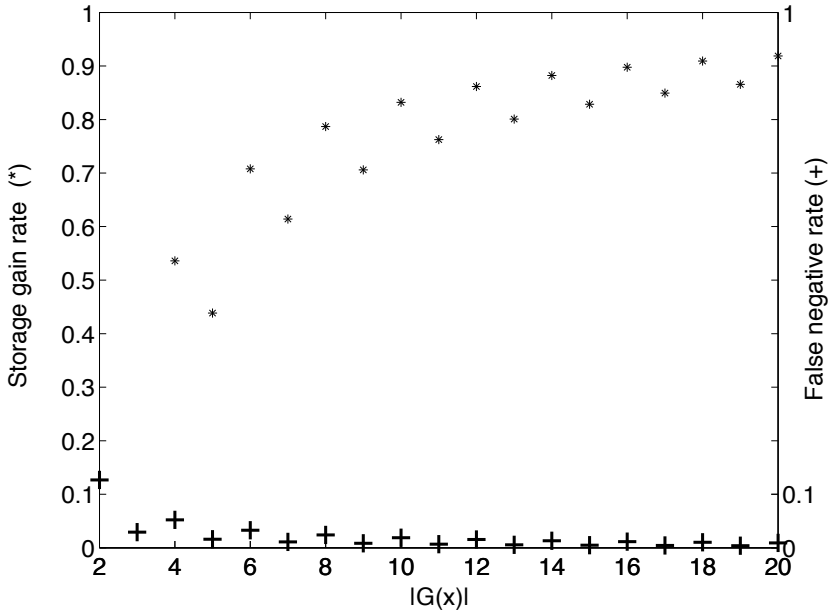


Figure 5.12: The variance of storage savings gain for a normal Gaussian distribution with $\delta_p = 0.5$ and its false negative rate as the number of neighbor SCRs increases.

focused on propagation of contexts to neighbors based on the graph inside a domain. Its drawback is that the property of a neighbor graph is based on geographical adjacency. Furthermore, it is not scalable since a single server must maintain the graph and relocate contexts among access points, which is difficult to extend to inter-domain transfers.

Extensive research has been done on mobility prediction for resource allocation in cellular networks. Liu *et al.* [51] introduced a hierarchical location prediction algorithm characterized by local and global prediction. In particular, the global prediction algorithm searches for one of user mobility patterns recorded in a user profile that is best matched with the user's actual path using approximate pattern matching, and selects the next cell inside the matched sequence. Its application to cross-domain mobility prediction is difficult, and requires knowledge of the entire multi-tier network topology. Instead of prediction of the next cell, Akyildiz and Wang [16] presented a method for predicting a zone towards

which mobile nodes will move; this is similar to ours, but differs in that our method offers geographical independence.

Trajectory prediction depends strongly on mobility models. The difference between different mobility models mainly lies in which of random elements, including speed, distance, angle, destination, and travel time, to choose and what probability distributions to use for each choice. We have considered a variety of mobility models discussed in [83].

5.7 Conclusions

By forming an overlay network, SCRs offer an attractive alternative to traditional mechanisms for transferring security contexts, especially in terms of flexibility, scalability, and deployability. In order to fully realize the SCR's potential, we must provide methods for transmitting (the replicas of) security context in a manner that is as scalable as the underlying consistency of the security context.

We overcame two challenges: (1) capture mobiles' direction patterns from the multi-tier access network and transfer security-context replicas via the overlay network, and (2) derive the efficacy of prediction from a combination of pattern matching and statistical estimation. Traditional prediction mechanisms lack transparent coordination among end-systems built with different technologies. Furthermore, straightforward flooding may at first appear effective, but suffers from its storage overhead. Our solution to predictive forwarding is shown to perform best while reflecting patterns in MNs' movement directions. It is robust to keep the false negative rate under 5%, even if the associations are distributed uniformly or randomly.

Our main contributions are efficient and scalable context management and algorithms that are effective in all circumstances. With these results, effective collaboration among

SCRs is enabled, and predictive forwarding of security contexts via the overlay network can be achieved.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

This thesis aims to advance the authentication technology for cross-domain mobility by addressing the requirements of both high-level security and latency-efficiency. The primary contribution consists in the development of a unified, scalable, high-performance authentication architecture that enables mobile nodes to seamlessly cross the domain boundary without compromising high-level security support. The architecture also supports transparency for access networks based on multiple wireless technologies. Our contributions are recapitulated as follows.

We first investigated a hierarchical authentication scheme, as opposed to the AAA architecture which is based on a client-server scheme. We introduced a deputy agent that plays a key role in enforcing the authentication policy on the basis of the security context that is derived from the mobile user's credentials. Furthermore, transferring the security context via interconnected deputy agents reduces long-distance message exchanges. The evaluation results demonstrated the effectiveness of the decentralized scheme in terms of

message traffic overhead.

Second, we proposed a mobility-adjusted authentication protocol, called MAP, that is dedicated to security context router (SCR) collaboration. MAP achieves considerable reduction in authentication latency in message exchanges. In addition, the feature of hierarchical key derivation makes the system more robust against various key comprising attacks. Experimental results and performance evaluation demonstrated that MAP takes full advantage of the concept of security context over other protocols favoring inter-domain authentication.

We then tailored an SCR with collaborative communications to form an overlay network. The key features of the SCR are twofold: it derives, replicates, updates and transfers the security context in a fault-tolerant way such that the existing security context is always available via SCRs, including the home server; predictive forwarding of the security context favors a perfect realization of seamless cross-domain mobility. Statistical estimation was applied to avoid false positive hits. Our numerical analysis and performance evaluation showed that these techniques are very effective and succeed in lowering storage overhead.

6.2 Recommendations for Future Research

As described above, our architecture augments the agility of secure cross-domain handoffs to play a key role in seamless mobility. To broaden the applicability of the architecture, nevertheless, there remain several open research problems associated with securing communications among SCRs.

- It is critical to make an architecture not only scalable and lightweight, but attack-tolerant as well. Each SCR is responsible for its domain and they connect to each other dynamically. To securely establish links and communicate with each

other requires authenticity and integrity of incoming messages, and privacy of security context. In this thesis, we focused on the security context management, but it is important to develop a mechanism to secure links among SCRs.

- It is impossible to predict the next domain to which a mobile node will move with 100% accuracy. We proposed a novel statistical approach to determine to which SCR the security context will be forwarded. In this thesis, this approach has been applied in coordination with the approximate pattern matching method, but other approaches could be adopted, e.g. the next associations could be determined by verifying membership using the Bloom filter data structure.

Another possible research direction would be to apply our proposed authentication architecture to other applications such as self-organized autonomous mobile devices.

6.3 Closing Remarks

Over the past few years, security concerns have been growing as wireless communications become pervasive. An authentication system plays a preliminary role in building a secure communication architecture. Conventional authentication systems based on a client-server scheme show limitations in cross-domain authentication in terms of scalability, efficiency and availability. Our cross-domain authentication architecture provides mobile users with secure seamless handoffs that are a prerequisite for secure multimedia services with acceptable QoS. The architecture is characterized by a decentralized scheme, a scalable authentication protocol, and security context management. We believe that the architecture presented in this thesis provides a good basis for a future secure wireless ubiquitous computing infrastructure.

Chapter 7

Acronyms

AAA Authentication Authorization Accounting

ACO Authenticated Cipher Offset

AES Advanced Encryption Standard

AS Authentication Server

AuC Authentication Center

AUTH AUTHenticator

AV Attribute Value

BAN Burrows, Abadi, and Needham

BNR Boundary Neighbor Register

BT Bluetooth Terminal

CAG Cellular Access Gateway

CAN Content-Addressable Network

CCMP Counter Mode with Cipher Block Chaining Message Authentication Code Protocol

DH Diffie-Hellman

DK Domain Key

EAP Extensible Authentication Protocol

GPRS Serving GPRS Support Node

GSM Global System for Mobile Communications

GTK Group Temporary Key

HLR Home Location Register

HMAC Hash Message Authentication Code

HNR *n*-Hop Neighbor Register

ID IDentity

IV Initialization Vector

KRT Key-associated Route Table

LAN Local Area Network

L2CAP Logical Link Control and Adaptation Protocol

MAC Message Authentication Code

MAP Mobility-adjusted Authentication Protocol

MIC Message Integrity Code

MN Mobile Node, interchangeable with mobile user

MNS Modified Needham-Schroeder symmetric-key protocol

NASREQ Network Access Server REQuirements

NS Needham-Schroeder symmetric-key protocol

PAN Personal Area Network

PDU Protocol Data Unit

PIN Personal Identification Number

PK Primary Key

PMK Pair-wise Master Key

PMKSA Pair-wise Master Key Security Association

PRF Pseudo Random Function

PTK Pair-wise Temporary Key

QoS Quality-of-Service

RAN Radio Access Network

RAND random number or random value

RC4 Rivest Cipher

RSN Robust Secure Network

RTT Round-Trip Time

SCR Security Context Router

SDP Service Discovery Protocol

SGT Service Granting Ticket

SIM Subscriber Identity Module

SRES challenge response value

STA mobile STAtion

TGS Ticket Granting Server

TGT Ticket Granting Ticket

TK Temporary Key

TLS Transport Level Security

TTL Time To Live

TVA Time-history Vectors of Angle

UMTS Universal Mobile Telecommunications System

USIM Universal Subscriber Identity Module

VLR Visitor Location Register

WPAN Wireless Personal Area Network

WEP Wired Equivalent Privacy

XRES expected response

Bibliography

[1] Authentication Authorization and Accounting IETF WG.

[2] Freenet. http://freenet.sourceforge.net/.

[3] Gnutella. http://gnutella.wego.com/.

[4] Napster. http://www.napster.com/.

[5] Wi-fi alliance. http://www.wi-fi.org/.

[6] Secure Hash Standard, Apr. 1995.

[7] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Specification for Robust Security, 1999.

[8] Advanced Encryption Standard (AES), Nov. 2001.

[9] Weaknesses in the key scheduling algorithm of RC4. volume 2259 of *Lecture Notes in Computer Science*, pages 1–24. Springer Verlag, 2001.

[10] 3GPP ts 33.102 3g security; security architecture. Technical Report v3.11.0, 3rd Generation Partnership Project, 2002.

[11] IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, Nov. 2003.

[12] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Specification for Robust Security, 2003.

[13] Bernard Aboba and Dan Simon. PPP EAP TLS Authentication Protocol. RFC 2716, Oct. 1999.

[14] William Aiello, Steven M. Bellovin, Matt Blaze, Ran Canettiand John Ioannidis, Angelos D. Keromytis, and Omer Reingold. Efficient, DoS-Resistant, Secure Key Exchange for Internet Protocols. In *Conf. on Computer and Comm. Security (CCS)*, Washington DC, USA, 2002. ACM.

[15] Ian Akyildiz and Wenye Wang. A dynamic location management scheme for next-generation multitier pcs systems. *Trans. on Wireless Comm.*, 1(1):178–189, Jan. 2002.

[16] Ian Akyildiz and Wenye Wang. The predictive user mobility profile framework for wireless multimedia networks. *Trans. on Networking*, 12(6):1021–1035, Dec. 2004.

[17] Juha Ala-Laurila, Jouni Mikkonen, and Jyri Rinnemaa. Wireless LAN access network architecture for mobile operators. *Comm. Magazine*, 39(11):82–89, Nov. 2001.

[18] Jari Arkko et al. Extensible Authentication Protocol Method for UMTS Authentication and Key Agreement (EAP-AKA). work in progress, IETF Draft.

[19] M. S. Bargh et al. Fast Authentication Methods for Handovers between IEEE 802.11 Wireless LANs. In *Int. Workshop on Wireless Mobile App. and Services on WLAN Hotspots (WMASH)*, pages 51–60. ACM, Oct. 2004.

[20] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols. In *30th Symposium on Theory of Computing*, pages 419–428. ACM, 1998.

[21] Larry J. Blunk and John R. Vollbrecht. PPP Extensible Authentication Protocol. RFC 2284, Mar. 1998.

[22] Nikita Borisov, Ian Glodberg, and David Wagner. Intercepting mobile communications: the insecurity of 802.11. In *MobiCom*, pages 180–189, Rome, Italy, July 2001. ACM.

[23] Torsten Braun and Hahnsang Kim. Efficient Authentication and Authorization of Mobile Users Based on Peer-to-Peer Network Mechanisms. In *38th Hawaii Int. Conf. on System Sciences (HICSS'05)*, page 306b, Big Island, Hawaii, Jan. 2005. IEEE.

[24] Michael Burrows, Martin Abadi, and Roger Needham. A Logic of Authentication. Technical Report 39, Digital Equipment Corporation, Palo Alto Calif., February 1989.

[25] Michael Burrows, Martin Abadi, and Roger Needham. A Logic of Authentication. *Trans on Comp. Systems*, 8(1):18–36, 1990.

[26] Pat R. Calhoun, John Loughney, Erik Guttman, Glen Zorn, and Jari Arkko. Diameter Base Protocol. RFC 3588, Sep. 2003.

[27] Pat R. Calhoun, Glen Zorn, David Spence, and David Mitton. Diameter Network Access Server Application. RFC 4005, Aug. 2005.

[28] Nancy Cam-Winget, Russ Housley, David Wagner, and Jesse Walker. Security flaws in 802.11 data link protocols. *Communications of the ACM*, 46(5):35–39, May 2003.

[29] Sunghyun Choi and Kang G. Shin. Predictive and adaptive bandwidth reservation for hand-offs in QoS-sensitive cellular networks. In *SIGCOMM*, pages 155–166, Vancouver, British Columbia, Sep. 1998. ACM.

[30] Wei Dai. Crypto++, http://www.eskimo.com/~weidai/ cryptlib.html.

[31] Tim Dierks, Alan O. Freier, Martin Abadi, Ran Canetti, Taher Elgamal, Anil R. Gangolli, Kipp E.B. Hickman, and Hugo Krawczyk. Transport Layer Security protocol version 1.0. RFC 2246, Jan. 1999.

[32] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. ISBN 0-471-05669-3. A Wiley-Interscience Publication, second edition, 2001.

[33] ETSI. Requirements and architectures for interworking between HIPERLAN/2 and 3rd generation cellular systems. Technical Report ETSI TR 101 957, ETSI, Sophia Antipolis, France, Aug. 2001.

[34] Paul Funk et al. EAP Tunneled TLS Authentication Protocol. work in progress, IETF Draft.

[35] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. ISBN 0-387-95284-5. Springer, 2001.

[36] Henry Haverinen and Joseph Salowey. EAP SIM Authentication. work in progress, IETF Draft.

[37] James Heather and Steve Schneider. Towards automatic verification of authentication protocols on an unbounded network. In *13th Computer Security Foundations Workshop*, page 132. IEEE, 2000.

[38] Markus Jakobsson and David Pointcheval. Mutual Authentication for Low-Power Mobile Devices. In *Financial Cryptography 2001*, Grand Cayman Island, British West Indies, Feb. 2001.

[39] Ping Ji, Zihui Ge, Jim Kurose, and Don Towsley. A comparison of hard-state and soft-state signaling protocols. In *SIGCOMM*, pages 251–262, Karlsruhe, Germany, Aug. 2003. ACM.

[40] Burt Kaliski. PKCS #1 RSA Encryption Version 1.5. RFC 2313, Mar. 1998.

[41] James Kempf. Context transfer problem statement. RFC 3374, Sep. 2002.

[42] Stephen Kent and Randall Atkinson. Security architecture for the internet protocol. RFC 2401, Nov. 1998.

[43] Hahnsang Kim and Hossam Afifi. Improving Mobile Authentication with New AAA Protocols. In *Int. Conf. on Comm. (ICC'03)*, Anchorage, USA, May 2003. IEEE.

[44] Hahnsang Kim, Hossam Afifi, and Masato Hayashi. EAP Bluetooth Application. work in progress, IETF Draft.

[45] Hahnsang Kim, Walid Ben-Ameur, and Hossam Afifi. Toward Efficient Mobile Authetication in Wireless Inter-domain. In *IEEE Workshop on App. and Services in Wireless Networks (ASWN'03)*, Berne, Switzerland, July 2003.

[46] Hahnsang Kim, Walid Dabbous, and Hossam Afifi. A bypassing security model for anonymous bluetooth peers. In *Wirelesscom*, volume 1, pages 310–315, Hawaii, U.S.A., June 2005. IEEE.

[47] Hahnsang Kim, Kang G. Shin, and Walid Dabbous. Improving cross-domain authentication over wireless local area networks. In *SecureComm*, pages 127–138, Athens, Greece, Sep. 2005. IEEE.

[48] John Kohl and B. Clifford Neuman. The kerberos network authentication service (v5). RFC 1510, Sep. 1993.

[49] Rajeev Koodli and Charles E. Perkins. Fast Handovers and Context Transfers in Mobile Networks. *SIGCOMM: CCR*, 31(5), 2001.

[50] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104, Feb. 1997.

[51] Tong Liu, Paramvir Bahl, and Imrich Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless atm networks. *JSAC*, 16(6):922–936, Aug. 1998.

[52] John Loughney, Madjid Nakhjiri, Charles Perkins, and Rajeev Koodli. Context transfer protocol. RFC 4067, July 2005.

[53] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1055, pages 147–166. LNCS, 1996.

[54] Paolo Maggi and Riccardo Sisto. Using spin to verify security properties of cryptographic protocols. In *9th SPIN Workshop on Model Checking of Software*, volume 2318, pages 187–204. LNCS, 2001.

[55] Catherine A. Meadows. Formal Verification of Cryptographic Protocols: A Survey. In *ASIACRYPT: Int. Conf. on the Theory and Application of Cryptology*, volume 917, pages 135–150. LNCS, Dec. 1994.

[56] Arunesh Mishra, Minho Shin, and William Arbaugh. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. *SIGCOMM: CCR*, 33(2):93–102, 2003.

[57] Arunesh Mishra, Minho Shin, and William Arbaugh. Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network. In *INFOCOM*, Hong Kong, Mar. 2004. IEEE.

[58] Arunesh Mishra, Minho Shin, and William Arbaugh. Pro-active Key Distribution using Neighbor Graphs. *Wireless Comm. Magazine*, 11(1):26–36, Feb. 2004.

[59] J. C. Mitchell, M. Mitchell, and U. Stern. Automated analysis of cryptographic protocols using murϕ. In *Symposium on Security and Privacy*, pages 141–153. IEEE, 1997.

[60] Michel Mouly and Marie B. Pautet. *The GSM System for Mobile Communications*. ISBN 2-9507190-07. Europe Media Duplication S.A, 1993.

[61] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Comm. of the ACM*, 21(12):993–999, 1978.

[62] Roger M. Needham and Michael D. Schroeder. Authentication Revisited. *SIGOPS: OSR*, 21(1):7, 1987.

[63] Clifford Neuman, Tom Yu, Sam Hartman, and Kenneth Raeburn. The kerberos network authentication service (V5). RFC 4120, July 2005.

[64] Dave Otway and Owen Rees. Efficient and timely mutual authentication. *SIGOPS: OSR*, 21(1):8–10, 1987.

[65] Sangheon Pack and Yanghee Choi. Pre-Authenticated Fast Handoff in a Public Wireless LAN based on IEEE 802.1x Model. In *IFIP TC6 Personal Wireless Comm. 2002*, Singapore, Oct. 2002.

[66] Sangheon Pack, Hakyung Jung, Taeyoung Kwon, and Yanghee Choi. SNC: A selective neighbor caching scheme for fast handoff in IEEE 802.11 wireless networks. *SIGMOBILE: MC2R*, 9(4):39–49, Oct 2005.

[67] Ashwin Palekar et al. Protected EAP Protocol (PEAP) Version 2. work in progress, IETF Draft.

[68] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. ISBN 0-07-366011-6. McGraw-Hill, fourth edition, 2002.

[69] Sylvia Ratnasamy, Paul Francis, Mark Handley, and Richard Karp. A scalable content-addressable network. In *SIGCOMM*, pages 161–172, San Diego, CA, Aug. 2001. ACM.

[70] Carl Rigney, Steve Willens, Allan C. Rubens, and William A. Simpson. Remote authentication dial in user service. RFC 2865, June 2000.

[71] Ronald L. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, Apr. 1992.

[72] L. Salgarelli. EAP SKE authentication and key exchange protocol. work in progress, IETF Draft.

[73] Apostolis K. Salkintzis, Chad Fors, and Rajesh Pazhyannur. WLAN-GPRS integration for next-generation mobile data networks. *Wireless Comm.*, 9(5):112–124, Oct. 2002.

[74] M. Satyanarayanan. Integrating security in a large distributed system. *Trans. on Comp. Systems*, 7(3):247–280, 1989.

[75] Minho Shin, Arunesh Mishra, and William Arbaugh. Improving the Latency of 802.11 hand-offs using Neighbor Graphs. In *Mobisys*, Boston, Jun. 2004. ACM.

[76] R. Shirdokar, J. Kabara, and P. Krishnamurthy. A QoS-based Indoor Wireless Data Network Design for VoIP. In *Vehicular Technology Conf. (VTC'01)*, volume 4. IEEE, Oct. 2001.

[77] Bluetooth SIG. Specification of the Bluetooth system, Core Version 1.1, Feb. 2001.

[78] Rodney Thayer, Naganand Doraswamy, and Rob Glenn. Ip security document roadmap. RFC 2411, Nov. 1998.

[79] Duncan S. Wong and Agnes H. Chan. Efficient and Mutually Authenticated Key Exchange for Low Power Computing Devices. In *ASIACRYPT: Int. Conf. on the Theory and Application of Cryptology and Information Security*, volume 2248, pages 272–289. LNCS, 2001.

[80] Duncan S. Wong and Agnes H. Chan. Mutual Authentication and Key Exchange for Low Power Wireless Communications. In *MILCOM 2001*, pages 39–43, USA, Oct. 2001. IEEE.

[81] Thomas Y. C. Woo and Simon S. Lam. A Lesson on Authentication Protocol Design. *SIGOPS: OSR*, 28(3):24–37, 1994.

[82] Chengshan Xiao, Karl D. Mann, and Jan C. Olivier. Mobile speed estimation for tdma-based hierarchical cellular systems. *Trans. on Veh. Tech.*, 50(4):981–991, Jul. 2001.

[83] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Sound mobility models. In *MobiCom*, pages 205–216, San Diego, California, Sep. 2003. ACM.

[84] Yuqing Zhang, Chunling Wang, Jianping Wu, and Xing Li. Using SMV for cryptographic protocol analysis: a case study. *SIGOPS: OSR*, 35(2):43–50, Apr. 2001.

FORMATION DOCTORALE

Dans le cadre de l'Ecole Doctorale SITEVRY
L'Institut National des Télécommunications en co-accréditation avec l'Université d'Evry - Val d'Essonne

(Cachet de l'établissement)

GROUPE DES ÉCOLES DES TÉLÉCOMMUNICATIONS
Institut National des Télécommunications
9, rue Charles Fourier
91011 EVRY Cedex

Fait à Evry, le 10 mai 2006

ATTESTATION DE DIPLÔME DE DOCTORAT

Je soussigné **Monsieur Christian MARGARIA**, Directeur Général de l'Institut National des Télécommunications, atteste que **Monsieur Hahnsang KIM** a soutenu sa thèse le **28 avril 2006** enregistrée sous le n° **06INT001**

Spécialité : **INFORMATIQUE**

Titre : « **Une architecture d'authentification pour la mobilité inter-domaine** »

Et qu'il (elle) a obtenu la mention « **Très honorable** »

Membres du Jury :

Monsieur Refik MOLVA
Madame Isabelle CHRISMENT
Madame Kaisa NYBERG
Monsieur Pascal URIEN
Monsieur Kang SHIN
Monsieur José ARAUJO
Monsieur Hossam AFIFI

Christian MARGARIA
Directeur Général de l'Institut National des Télécommunications

Le diplôme définitif sera envoyé ultérieurement

TELECOM & Management SudParis

Dans le cadre de l'Ecole Doctorale SITEVRY
En co-accréditation avec l'Université d'Evry Val d'Essonne

(cachet de l'établissement)
Institut TELECOM
TELECOM & Management SudParis
9 rue Charles Fourier
91011 Évry Cedex

Evry, 12 December 2008

DOCTOR DEGREE ATTESTATION

I, undersigned **François SIMON**, Director of Research and Doctoral studies of TELECOM & Management SudParis (ex Institut National des Télécommunications) confirms that **Mr.KIM Hahnsang** received a Doctor Degree in **«Computer Science»** on the 28th April 2006. The thesis is registered under number. **06INT001**.

Title : **« Une architecture d'authentification pour la mobilité inter-domaine »**

He has obtained the highest honors.

François SIMON
Directeur de la Recherche
et des Formations Doctorales

François SIMON
Director of Research and Doctoral studies of TELECOM & Management SudParis

9 rue Charles Fourier
91011 Évry Cedex
France
Tél : + 33 (0)1 60 76 40 40
Fax : + 33 (0)1 60 76 43 25
Siret : 180 092 025 00048
APE : 803Z
www.it-sudparis.eu