RIKU ITÄPURO

SMARTPHONE AS A TRUST ANCHOR IN HOME NETWORKS

draft-20.8.2015 Master of Science thesis

TERMINOLOGY

If not already on vocabulary, expansion of the most important terms like authentication, key-exchange, integrity, replay, algorithms, SIM,... [from Cryptoprotocol-course, check that key exchange with 8 different methods)]

RIKU ITÄPURO

SMARTPHONE AS A TRUST ANCHOR IN HOME NETWORKS

draft-20.8.2015 Master of Science thesis

Examiner: Prof. Jarmo Harju
Examiner and topic approved by the
Faculty Council of the Faculty of
Computing and Electrical Engineering
on 4th February 2015

ABSTRACT

RIKU ITÄPURO: Smartphone as a trust anchor in home networks
Tampere University of Technology
draft-20.8.2015 Master of Science thesis, xx pages, x Appendix pages
xxxxxx 2015
Master's Degree Programme in Information Technology
Major: Information Security
Examiner: Prof. Jarmo Harju
Keywords: authentication, authorization, AAA, homenet, smartphone, SIM, trust-anchor, EAP-SIM, RADIUS

[what was the problem, what was done, and what are the results.]

Today, home networks have become more complex and home owner does not necessarily want to administer all aspects of it. Configuring home network devices does not differ much from configuring enterprise devices. One needs access, credentials to login and knowledge to operate the device. If configuration is out-sourced to external parties and done remotely, those requirements need adjustment. Access to the end device must be provided from outside. Login credentials must be shared to operator and trustful operator must be hired to make configuration changes. For that, some beforehand set provisioning and distribution of authentication keys is needed. As there already exists an infrastructure within mobile phone subscribers, that is used in the study as a trusted base. To benefit from mobile identification it is shown how authentication is done using extendable authentication profile (EAP) with SIM-card and authorization checked with RADIUS protocol. A theory, how SIM-authentication works is presented and a simulated environment to demonstrate that is built, tested and analyzed. As a result it is shown, that SIM authentication's benefits are strong authentication and existing user-base, while its disadvantages include dependency to mobile operator. Additionally, there will remain challenges in keeping SIM's identity private and in disabling unwanted re-authentications. Principle has been to reuse existing techniques when combining them to such new areas as homenet and delegated management. For transporting authentication claims, WPA2 enterprise has been chosen, which includes RADIUS environment. To further avoid complexity and granularity, we only use a simple model of management network. Getting in to management network is carried out at home network via EAP-SIM authentication and it is the key element of the thesis.

TIIVISTELMÄ

RIKU ITÄPURO: Älypuhelin kotiverkkojen luottamusankkurina
Tampereen teknillinen yliopisto
Diplomityö, xx sivua, x liitesivua
toukokuu 2015
Tietotekniikan koulutusohjelma
Pääaine: tietoturva
Tarkastaja: Prof. Jarmo Harju
Avainsanat: tunnistaminen, valtuutus, AAA, kotiverkko, älypuhelin, luottamusankkuri, EAP-SIM, RADIUS

The abstract in Finnish. Foreign students do not need this page. TBD

Kirjoita, kun english versio on hyvä(ksytty).

PREFACE

PREFACE TEMPLATE! SKIP.

This document template conforms to Guide to Writing a Thesis at Tampere University of Technology (2014) and is based on the previous template. The main purpose is to show how the theses are formatted using LaTeX (or LaTeX to be extra fancy) .

The thesis text is written into file `d_tyo.tex`, whereas `tutthesis.cls` contains the formatting instructions. Both files include lots of comments (start with %) that should help in using LaTeX. TUT specific formatting is done by additional settings on top of the original `report.cls` class file. This example needs few additional files: TUT logo, example figure, example code, as well as example bibliography and its formatting (`.bst`) An example makefile is provided for those preferring command line. You are encouraged to comment your work and to keep the length of lines moderate, e.g. <80 characters. In Emacs, you can use `Alt-Q` to break long lines in a paragraph and `Tab` to indent commands (e.g. inside figure and table environments). Moreover, tex files are well suited for versioning systems, such as Subversion or Git.

Acknowledgements to those who contributed to the thesis are generally presented in the preface. It is not appropriate to criticize anyone in the preface, even though the preface will not affect your grade. The preface must fit on one page. Add the date, after which you have not made any revisions to the text, at the end of the preface.

Tampere, 1.5.2015

Teemu Teekkari

TABLE OF CONTENTS

1. Introduction 1

2. Authentication, Authorization, and Trust 5

2.1 802.1X 5

2.2 RADIUS 6

2.3 WPA2 7

2.4 EAP 8

2.5 SIM-based authentication 8

2.6 Trust 12

2.6.1 to somewhere else, is not trust property 13

3. Managing Home Networks [or Home network architecture] 14

3.1 Home network architecture and IETF 14

3.2 Centralization trends in management 16

4. Design of home network trust anchor and separation of change management 17

4.1 Alternative methods for introducing trust anchor into the homenet . . 17

4.2 Flow of design [TBD already above, FIX THIS section] 20

4.3 Chosen design and why (Rationale) 21

4.4 [Need for Security bootstrapping] 23

4.5 Access methods to Wi-Fi with only one SSID 24

4.5.1 HS2.0 [If deleted, remember also from conclusion! TBD] 24

4.6 Scenarios for choosing the AuthZ location 25

4.7 Ways to modify RADIUS messages [perhaps to security integrity chapter?] 29

4.8 Similarities with Lock-and-Key method 31

4.9 Summary of the chosen solution] 32

5. Implemented Solution 34

5.1 EAP-SIM authentication test bed 34

5.2 Detailed description of test runs 35

5.3 Disconnecting the local controller and offline changes 36
5.4 Network traces (EAP, SIM, AUTH traffic analysis) 37
6. Analysis, Results and Discussion 40
6.1 Deployment difficulty 40
6.2 Estimating time to authenticate EAP-SIM [TO BE REMOVED] . . . 40
6.3 Costs for end-user 40
6.4 Platform specific issues 41
6.5 Security considerations 42
6.5.1 Confidentiality (privacy) 42
6.5.2 Integrity 44
6.5.3 Accessibility, DoS and Scalability 44
6.5.4 RADIUS weaknesses and strengths in limited use cases 45
6.5.5 Replay, Re-use, Re-auth, and brute-force challenges 45
6.5.6 Mitigation methods 46
6.5.7 Decision point for adding role information [move to design part] 47
6.6 Discussion 48
7. Conclusion 50
Bibliography 55
APPENDIX A. Scripts, confs, and logs 56
A.1 shell, logging options 56
A.2 wpa-supplicant settings (wpa-simtest-owrt2.conf) 58
A.3 RADIUS server conf 59
A.4 hlr auc 59
A.5 No sim 59
B. [MISC to be added on right places] 60
B.1 facts TBD. 60
B.2 using EAP for other than network access, i.e., for application auth. . 61
B.3 eap-psk rfc4764.txt 61

B.4 eap-sim acts similar than any other EAP challenge method (or not?) 61

LIST OF FIGURES

1.1 Local Controller and Collaborative Management Design 2

2.1 EAP-logical layering 8

2.2 Simplified EAP-SIM sequence diagram, source RFC4186 11

2.3 EAP-SIM full authentication with RADIUS 12

4.1 Scenario I with 3 separate domains: BaaS, MNO and homenet 26

4.2 Scenario II with AuthZ in homenet 28

4.3 Scenario III with outsourced AA 29

4.4 Scenario IV, AuthZ from BaaS, AuthN from homenet 29

4.5 802.1x access control with EAP [TBD deleted?]. Source Cisco. 31

4.6 Cisco's view of Lock-and-key access 31

5.1 EAP-SIM AuthN messaging in simulation testbed 34

6.1 Using TMSI 43

LIST OF TABLES

2.1 Comparison of WPA2-PSK and WPA2-ENTERPRISE modes 7

2.2 Setup tasks for clients in WPA2-Enterprise with EAP-PEAP-MSCHAPv2 and EAP-SIM 11

4.1 Location of AA, AuthN and AuthZ in scenarios I-V 26

LIST OF PROGRAMS

./testit/apd–tty.clean 56

testit/wpa–simtest–owrt2.conf.clean 58

testit/hostapd–jmdemo.conf.clean 59

LIST OF ABBREVIATIONS AND SYMBOLS

TUT	Tampere University of Technology
URL	Uniform Resource Locator
3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization, Accounting
AKA	Authentication and Key Agreement
AuC	Authentication Center
CPE	Customer Premise Equipment
EAP	Extensible Authentication Protocol
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
GSM	Global System for Mobile Communication (earlier Groupe Spécial Mobile)
HLR	Home Location Registry, ...
IEEE	Institute of Electrical and Electronics Engineers
IMSI	International Mobile Subscriber Identity
ISP	internet service provider
MNO	mobile network operator, owner of cellular network, knows SIM secrets
RADIUS	Remote Authentication Dial In User Service, protocol and server, AAA service
SIM	Subscriber Identity Module, a smartcard. Also USIM program running in UICC card (UMTS networks)
SSID	Service Set Identifier, identifies Wi-Fi network
TMSI	Temporal Mobile Subscriber Identity
Wi-Fi	Wireless local network, implements IEEE 802.11 standards
WPA	Wireless Protected Access version 1.
WPA2	Wireless Protected Access version 2, more secure than WPA

802.1X port based access control standard

Access point Wi-Fi client connects access point (AP) on 802.11 layer. AP knows EAP client and encapsulates EAP-message to RADIUS-message and forwards that to Authenticator.

Authenticator local entity, who makes authentication (and authorization) decision for client based on local and remote claims, part of 802.1X standard.

mobile-operator MNO, knows connection between SIM-owner and SIM

proxying RADIUS RADIUS server standing between RADIUS client and Authentication server, part of RADIUS server chain.

1. INTRODUCTION

Managing computer and network devices can be hard. Modern homes have become similar to small offices regarding the equipment present there. Earlier it was sufficient to make just minimal settings at home to a modem (cable, phone or radio) and connect it to the home computer to get a fully working home network with internet connectivity. Now home network has expanded with countless devices available. Already entertainment centers (AV-amplifiers, media players, gameconsoles), manageable network devices (switches/routers), and mobile phones present new devices and network segments beside computers and printers. Sensors and controller devices from Internet of Things domain bring their own increment to the device count at home. Connecting these devices to the net remains trivial, but managing the network afterwards has become challenging and complex.

There might be separate areas in homes that have different needs regarding connectivity, resources, and access. Not only that, but devices in separate segments might not belong to the home owner anymore, hence needing their own administrative parties. For example an electricity company may have a sensor and controller network, which physically uses the home network, but logical is separated from other parts of home network. It is then important to keep track of who is allowed to access which part of the home network.

The configuration choices in networking devices takes some amount of expertise what is not necessarily present at every home. There could exist a market for an external consultant service, which would remotely manage the home network. Remote management eliminates consultant's physical precence at home and so reduces external actors' costs, but adds many questions regarding security, not only to overcome firewalls, NATs, and disconnections. Persons, who are allowed to make configuration changes, are today often authenticated only by simple password and physical precence at home. If external help were present, home owner would need more control allowing only authorized operators in, because earlier used physical precence protection is missing.

Lastly, it is challenging to find a common, trusted entity, upon which every actor could base their trust. Common trust is needed to join before unknown parties together safely. Summarized the problems are delegated network management, remote provisioning, and trust finding. On its roots, this is an authentication and authorization problem.

One model to solve these problems is to separate the management and control function away from the connectivity and routing issues. Silverajan et al.[35] proposes a model where the management is achieved through a service in a cloud. The cloud service is of type Backend as-a-Service (BaaS) which is suited well for mobile app usage. The configuration model of devices at a home is mirrored to the cloud as a resource graph (Figure x.1 TBD). Changes can be planned ahead at cloud and committed (Pushed) later to the home using configuration tools CoAP and RESTCONF via a local controller point which lies at home.

The managers operate in the cloud with the configuration data and the cloud verifies those managers, but the question remains, how to connect the cloud service to the home network through the local controller securely. The local controller at home would approve the changes and a smartphone is assumed to function in local controller role. It will operate as a bridge between the cloud and the home network. See Figure 1.1 for design of this architecture.

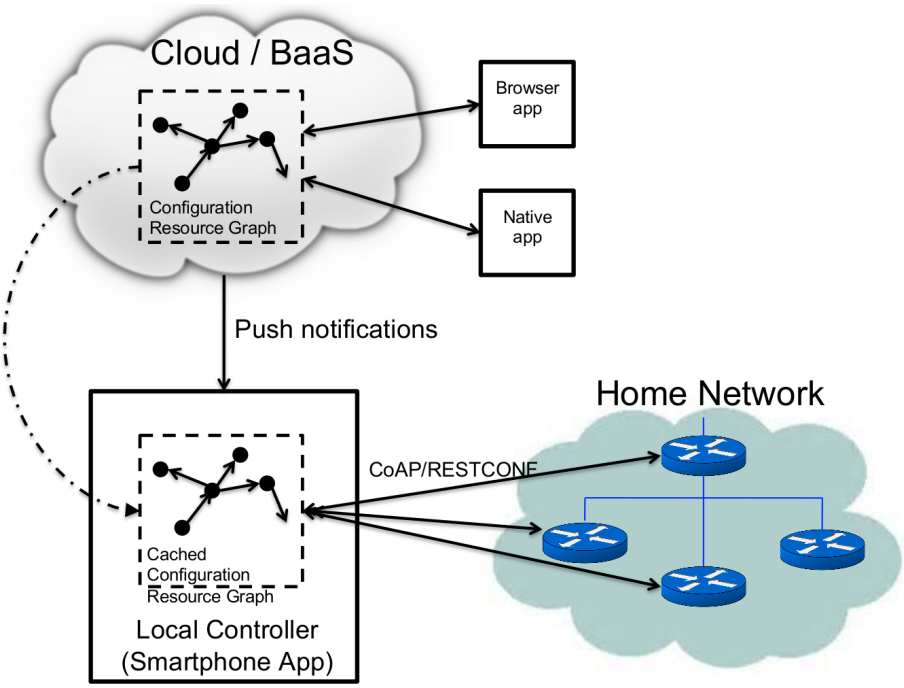

Figure 1.1 *Local Controller and Collaborative Management Design*

The delegated service provider therefore does not need to have a direct data access to the home but only to the cloud based service in order to be able to manage home

network devices. Consultant service is not the only possible delegation for home network.

One of the security issues are the authentication and authorization from the cloud to the home net. To secure the connection from the cloud service (controller) to the home network, there needs to be a common trust between each end point and the research problem here is how to enable the trust between the controller and the home network as the controller lies at the edge of the home network.

Any encryption between devices needs trusted key exchange beforehand, and finding and establishing trust is needed for that. That is called the key distribution problem. Public and privat keys solve the key exchange part, but only partially, because the trust still must be found somewhere. Above mentioned cloud solution for delegated home network management currently has preliminary authentication and access model using pre-defined credentials and SSH-connection from the local controller device to configuration targets[35, Chap.4]. That does not yet handle the bootstrap of the infrastructure i.e. the first trust is thought as given.

The smartphone with its Subscriber Identity Module (later SIM) and an existing key infrastructure to mobile network operator (MNO) would later eliminate the requirement for an additional credential distribution and that issue is studied in this thesis. Although the smartphone provides alternative authentication method with its SIM key, usual methods to authenticate still are plain username-password combinations. Those security issues must be solved before delegation in the cloud can happen.

The trust can be derived from the facts that already are known. The place, where a trust is not anymore derived or built upon other fact but assumed to be present, is called a trust anchor. [OR: *The trust anchor is then the fact, state or place, where derivation is done no more, but accepted per se.*] Combining existing techniques, this thesis presents one possible way to bind home network's trust to the smartphone's unique, existing secret keys inside the smart card's Subscriber Identify Module (SIM), which then would function as a trust anchor. To generally find ultimate trust it is only needed to verify trust chains until the chain reaches a trust anchor.

Human aspect and usability are important, although the focus will still be on authentication and authorization part of the home net management with smartphone as trust anchor. The proposed model should nevertheless require less effort than the current used methods on distributing user credentials, finding right place for them

to be inserted, and ensuring that they are written correctly. Besides those, problems such as limited connectivity are studied.

The thesis is structured as follows: in Chapter 2 the authentication–authorization model is explained. Chapter 3 describes security in current home net architecture and current practices for configuring it. Chapter 4 discusses methods to bring a trust anchor in the home network and explains the chosen method. One specially crafted problem is how the scenarios presented here can be tested without knowing SIM card's secret keys and without real phone operator involved. Those experiments are described in Chapter 5. Results are discussed on Chapter 6 and Chapter 7 concludes the thesis.

2. AUTHENTICATION, AUTHORIZATION, AND TRUST

[TBD? 4) Feature comparison, eg role-based access, time-based access etc]

Authentication, authorization, and accounting services (AAA) are components for access management. AAA-protocols do not dictate policies, i.e., who is granted access or what operations user is allowed to do. They only transport this information between client who needs them and server authorized to provide them. Often, the last 'A' which stands for accounting has been neglected and also here only first two A's are used and later described as AA services. Authentication (AuthN) answers how to identify users and prove that they really are who they claim to be. Authorization (AuthZ) answers what operations the identified users are allowed to do and forces usage policy. The rest of the thesis uses shortened terms AuthN and AuthZ.

On very small environments AA service is built on static backend such as file on protected target that the object wants to access. There AuthN is checked against a credentials file and authorization from a service specific policy file. To be more exact, identification preceding authentication is the part, where entity claims and presents its identity to access controlling system. That can involve sending username, login name or other identifier. Authentication in turn is the part where those facts are verified. AuthZ involves checking, which rights are available for authenticated entity.

Before we introduce SIM-based authentication used throughout the thesis, protocols 802.1X, WPA2, EAP and RADIUS are described in the following Sections. [sections?]. Last, we expand the term trust.

2.1 802.1X

802.1X[17] is an IEEE standard protocol for port based access control. Ports are physical layer ports, not to be mixed to Layer-4 ports such as TCP/UDP ports. Network access through a specific physical port is restricted (controlled) from a client (called Supplicant) before the client has successfully performed an AA. A

802.1X device, where the ports are located, is called the Authenticator. Third party in 802.1X is an Authentication server.

It is easy to mix here terms *Authenticator* and *Authentication server*, but their roles are different: Authenticator works as a gate-keeper to ports between supplicant and network, while Authentication server handles AA processes. At home, Authenticator usually lies inside the access point, but on large enterprise networks, Authenticator can be a centralized unit and multiple access points function only as radio stations.

2.2 RADIUS

RADIUS is the most popular provider for AAA-services[11, p.75]. It was used first with remote terminal and dial-up modem users, hence the name Remote Authentication Dial-In User Service. Later it was used as centralized AAA for networking devices such as switches and routers.

RADIUS protocol is a stateless, request-response type client-server protocol. There are four types of RADIUS messages defined in RFC 2865 that are used in AA. ACCESS-REQUEST and ACCESS-CHALLENGE cover both AuthN and AuthZ messaging while final RADIUS message is either ACCESS-ACCEPT or ACCESS-REJECT, based on the result given by the RADIUS authentication server.

Today, RADIUS has some shortcomings and fixing them is not anymore reasonable as developing has shifted to another AAA protocol called Diameter, which is already in use in 3GPP and 4G networks[38]. Nevertheless, as RADIUS is so wide-spread, it is still used in lots of places instead of Diameter. Currently, the main environment of RADIUS, besides network managing, is wireless connections (Wi-Fi) in enterprises and nationwide community federations.

When local Wi-Fi groups in Finland such as “SparkNet”, “Langaton Tampere”, or “Wippies” started to form in around 2005, they used 802.1X and RADIUS for AA. Those networks did still have as an alternative AA method a captive portal technique, where user first authenticates on WWW-page before getting an access. 802.1X and RADIUS enabled users in those group to consult external, central RADIUS server for authentication requests automatically, without burden that captive portal brings. The members of Wi-Fi groups can use network anywhere, where the same uniform SSID (Service Set IDentifier) was seen, i.e., roaming became possible, if one found a familiar SSID outside home area. Later, there were agreements between different local groups to allow roaming and so federations were born.

As seen from federated Wi-Fi groups, RADIUS servers can be chained to form a tree. The reasons for the chaining are load balancing and high availability, centralization of locally distant servers, and federation of different domains. In RADIUS trees, the messages are chained and proxied to next RADIUS server, depending on the settings on the proxying RADIUS server.

In the following chapters it is discussed how proxying servers take part in AA decisions. Of main interest there is, if it is possible to inject or modify AuthZ information in those proxying RADIUSes in cases, where AuthN and AuthZ are provided from different places[2]. Secondary goal is to universally divide AA regarding clients domain in the federation.

2.3 WPA2

Wireless protected access (WPA or WPA2) protects traffic in wireless, shared media, where everyone otherwise can simple listen all the radio traffic. It enables both authenticated access and message encryption between the smartphone and wireless access point (AP) by negotiating session keys after 802.1X has opened the virtual port in AP for the Supplicant.

WPA (version 1) was an early subset of then upcoming 802.11i standard, while WPA2 is the full implementation, also noted as IEEE 802.11i-2004, and the term WPA2 is used throughout the thesis. Client software for 802.11i is called a WPA2-Supplicant and it is used in wireless clients to communicate with the Authenticator.

WPA2 has two modes of protection: one for groups with common, pre-shared key (WPA2-PSK, also known as WPA2-Personal) and one for individuals (WPA2-RADIUS, also known as WPA2-Enterprise). With WPA2-RADIUS, revoking individual access is easier, but client setup slightly more complicated than on WPA2-PSK, as seen on Table 2.1.

Table 2.1 *Comparison of WPA2-PSK and WPA2-ENTERPRISE modes*

Property	WPA2-PSK	WPA2-ENTERPRISE
for groups	x	
for individual		x
client setup	easy	intermediate
individual client revocation		x

2.4 EAP

Instead of bringing new AuthN methods into 802.1X, it was extended with a modular framework called EAP (Extensible Authentication Protocol)[1]. Researchers justify using EAP, as it provides flexibility independent from underlying technology, whether wireless or wired, and integration with AAA infrastructures, although it adds some amount of time for authentication.[31]. Different authentication methods, for example hashed passwords, TLS certificates, or SIM/AKA using smartphone's SIM card, can be used with EAP. This work uses EAP-SIM authentication method.

EAP describes only the messaging form, so EAP messages needs to be encapsulated inside another protocol. In Wi-Fi, between a smartphone and an AP, EAP is encapsulated into 802.1X protocol (as EAPOL) or into protected EAP(PEAP)[29] before sending into air. In wired net those EAP messages are translated and encapsulated into RADIUS.

The encapsulation is described in Figure 2.1 where it can be seen, that EAP messaging happens logically between the EAP peer and the Authentication server. On a lower transport layer there is an EAP Authenticator in between them, which transfers EAPOL messaging into RADIUS message.

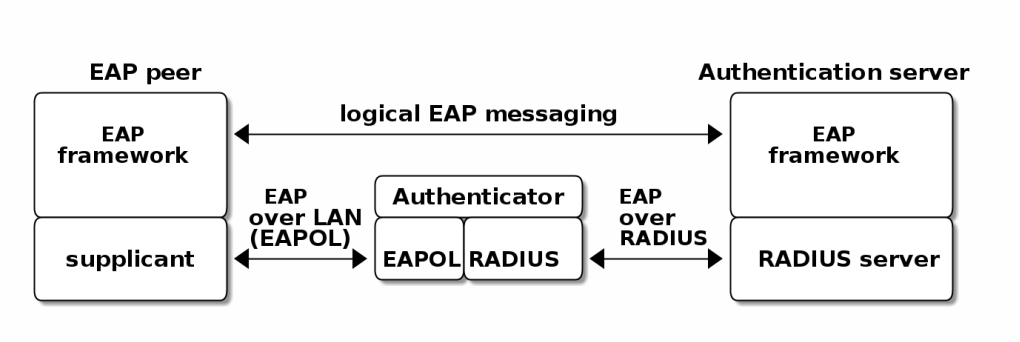

Figure 2.1 *EAP-logical layering*

Further, EAP is used to transfer AuthN messages only. It included neither AuthZ information, which is RADIUS's responsibility nor session keys, which are negotiated by WPA2. In the end, the Authenticator is the responsible for opening access for EAP peer as 802.1x dictates.

2.5 SIM-based authentication

SIM associates a physical card used in smartphones to a subscriber of the Mobile Network Operator (MNO). SIM here means the secret keys and the application in

mobile phone's SIM or USIM inside UICC(Universal Integrated Circuit Card). The secret keys are hardware protected and only usable to applications in SIM card. The SIM's storage also includes a unique serial number ICCID (Integrated Circuit Card Identifier) which identifies SIM globally and a unique IMSI (International Mobile Subscriber Identity). IMSI is a composition of digits belonging to Mobile Country Code(MCC, 2 digits), Mobile Network Code(MNC,2-3 digits) and Mobile Subscriber Identification Number(MSIN, 10 digits at most). More familiar, it is the user's phone number. [More detailed SIM architecture? Java Card, CPU,?]

SIM card usage can be controlled by two passwords: PIN and PUK. PUK is used as a remedy, if PIN has been inserted wrong too many times. If the card has other applications, for example mobile electrical signature application Mobiilivarmenne, they may have different keys and codes.

MNO distributes SIM card and provides mobile network connectivity to its customer. The secret keys are used for authenticating the IMSI to MNO and that enables MNOs to identify their customer in the network and charge them correspondingly. It is assumed, that SIM card represent its owner but in reality nothing prevents an identity thief to steal someone's SIM card. Although 4-digit PIN tries to prevent the usage of stolen SIM, that is considered as a weak safe[25, p.31].

AA services need to trust some entity endpoint and in case of MNO and SIM, they already mutually trust each other, and SIM can be used to open access to mobile networks. Separate access credentials for Wi-Fi access were still needed and that was the reason of developing EAP-SIM and later the derivatives EAP-AKA and EAP-AKA'. The goal was to combine in a secure way existing keys used in GSM (Global system for Mobile communication) Wi-Fi access. Existing general purpose EAP-methods in 2004 were not compatible with GSM protocols for this purpose.[16, p.93] The result of development was that today SIM can be used via EAP-types EAP-SIM, EAP-AKA, or EAP-AKA'(AKA-PRIME).

EAP-SIM is the original type created for GSM networks and defined in RFC4186[15]. It is a challenge-response method and similar to AuthN used in GSM, but it adds mutual AuthN, i.e., also the network is authenticated. Network authentication is achieved, if network is able to correctly respond to a client sent nonce, which by definition is a value used only once. The nonce can be thought as client's challenge to network.

The client in turn is authenticated, when the Authentication server generates a challenge with aid of triplet from MNO and the client responses to it correctly. That procedure is later described in more detail.

Beginning from 3GPP networks, new types EAP-AKA and AKA' can be used. EAP-AKA is defined in RFC4187[4] and uses 3GPP's AKA (Authentication and Key Agreement) protocol. It differs from EAP-SIM by using additionally parameters such as sequence numbering from MNO to protect replay attacks. Additionally, digests use SHA-256 function instead of SHA-1.[5]. Otherwise the protocol messaging is same as in GSM-SIM, only algorithms differ. Last, there exists EAP-AKA' that enhances AKA by including Service Set Identifier (SSID) in the key derivation function, which limits the possibility of using possibly compromised network's nodes and keys.

Using EAP-SIM means using the secret key inside SIM card with A3/A8 algorithms to generate valid responses for challenges coming from MNO and to derive session keys. The algorithms used (A3/A8) and their possible implementations (COMP128, COMP128v2, COMPv3) are not of interest in this work beside the point that they are MNO specific or known reference algorithms.

Disadvantages with SIM is dependency on mobile operator and internet connection, although disconnectivity issues are later addressed partly. Using smartphone may cost money, either to client or to service provider, although costs could be lower than using SMS, because IP network is used instead of mobile phone network.

In many parts, SIM variants in EAP are simpler, than other EAP variants to mobile client. Table 2.2 compares the setup of Wi-Fi in clients of one existing organization compared to EAP-SIM. It is noteworthy, that plain EAP-SIM will not support identity hiding and that will be later discussed further. If we added PEAP also to EAP-SIM (in last column of Table 2.2), comparison would be more fair. As can be seen from the table, leaving certificates out from the environment makes client setup easier with the price of revealing smartphone user's identity.

Bird's eye view to the EAP-SIM messaging between the smartphone, AP, Authentication server and MNO is described in Figure 2.2. The traffic is EAP on the left, RADIUS in the middle, and MAP/SS7, which is cellular telephony application, on the right.

Important parameters for this work are IMSI, NONCE, and triplet values corresponding IMSI (RAND, SRES, Kc). Sequence diagram of full EAP-SIM authentication between Supplicant (here smartphone) and Authenticator (in AP) is shown in Figure 2.3. There we can see that IMSI is used in message 2. IMSI is the identity, which Authentication server would next try to challenge and for which the AuthZ would be checked.

Table 2.2 *Setup tasks for clients in WPA2-Enterprise with EAP-PEAP-MSCHAPv2 and EAP-SIM*

Task: (x)="needed", (N/A)= "not available"	EAP-PEAP with MSCHAPv2	EAP-SIM	EAP-PEAP with EAP-SIM
CA settings:			
- choose CA for the RADIUS	x		x
- tell CA to clients	x		x
- if CA not known, distribute it *securely*	x		x
Other settings:			
- set used EAP-method	x	x	x
- set validation of RADIUS server's name	x		x
- set encapsulation (WPA2/802.1X)	x		
- set password	x	x(PIN)	
Identity hiding:			
- enable PEAP	x	N/A	x
- set outer identity	x	N/A	x
- set inner identity	x	N/A	

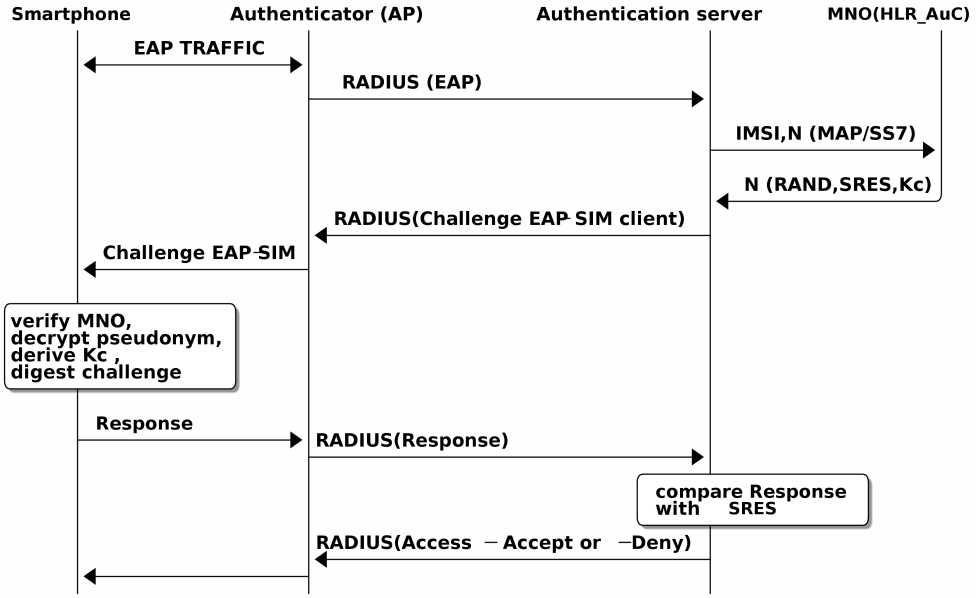

Figure 2.2 *Simplified EAP-SIM sequence diagram, source RFC4186*

All EAP-SIM derivatives provide mutual authentication. An operator is authenticated, when the client challenges it by sending a NONCE value during the start of the negotiation phase in the message 4 (Figure 2.3). The client later checks in the process 7, whether RAND values from the operator were digested with the correct NONCE.

After session has been set, IMSI may be left out and a temporal IMSI (TMSI) can be used instead to hide client's identity, for example in fast re-authentication case to reduce the risk of exposing the client's IMSI unnecessarily. Unfortunately, at that point, IMSI has already been exposed once in plain text, namely in message 2. It must be noted, that TMSI used here differs from TMSI used in 3GPP networks. Those context must not be mixed, otherwise the security that they bring may reduce, i.e. one must not use the TMSI received from 3GPP as TMSI in EAP-SIM.

TMSI is composed of pseudonym and realm part and can be a string. So, one can send `my-string-which-can-change@...operator.domain` instead of IMSI number (or `IMSI@.....operator.domain`) as an identity.

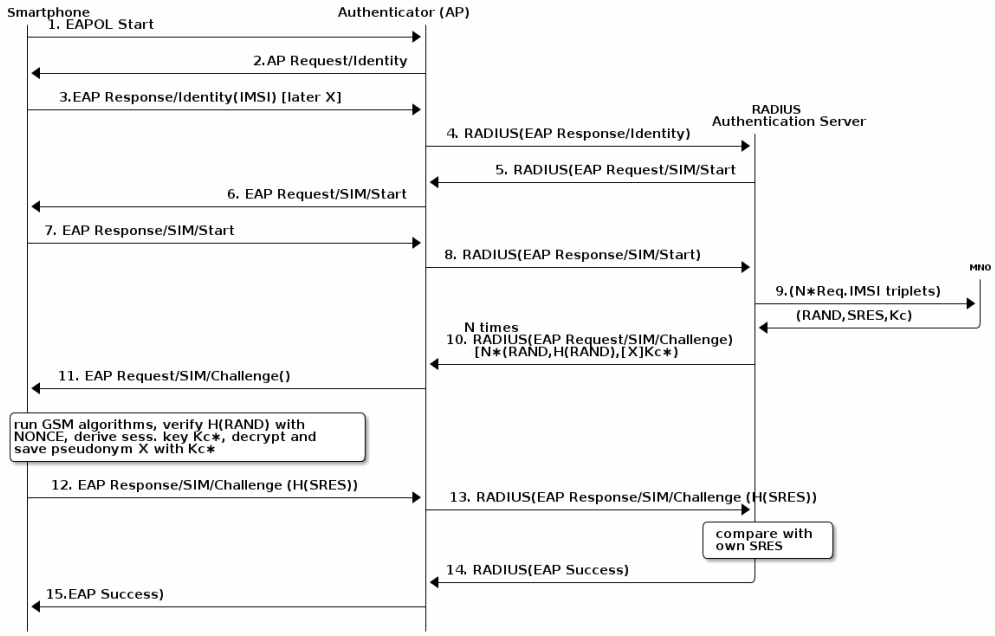

Figure 2.3 *EAP-SIM full authentication with RADIUS*

2.6 Trust

Secure communication has many layers and on its base lies trust. Only after completing trust setting phase it is meaningful to complete the other security layers. For example, secret keys enable encrypted communication, but they need to be delivered first through an trusted channel. Same applies to public key infrastructure solution, when exchanging public parts of keys and so it can be seen that trust really is the first layer to be fixed.

Even without trust, some form of secure asymmetric key-exchange is achievable with Diffie-Hellman key-exchange[10]. Unfortunately, it is vulnerable to Man-in-the-

middle(MitM) attacks, where protocol does not notice, if messaging goes through third party, which impersonates itself to both ends as being the corresponding messaging partner. MitM can read and decrypt encrypted messages and forward possibly changed message with correct looking signature. With trust set between two devices, i.e., if they can securely authenticate each other, secret communication is possible. Secure network configuration and credential exchange is then possible.

Now, how this trust could be used to include other components under same trust circle in the home network? As mentioned earlier, the SIM and MNO trust each other hence mutual authentication between them is possible and that is later shown to be an important factor. Also the key distribution problem mentioned in Chapter1 is solved already at SIM-card distribution phase. As AuthN-AuthZ at home proceeds through the Authenticator, maybe the Authenticator can deliver this information further and use it as a derivation function to extend trust.

2.6.1 to somewhere else, is not trust property

EAP-SIM derivatives provide strong AuthN which means here two-factor AuthN. Software certificates, while stronger than regular passwords, on the other hand do not possess the properties *non-copiable* or *unique*, so they can only be considered as strong passwords and they do not full-fill requirement for two-factor AuthN. If we nonetheless were using software certificates with method such as EAP-TLS, then the certificates (for CA and client) and the private key should still be provisioned first, which would defeat what we wanted to achieve in easy user experience.

3. MANAGING HOME NETWORKS [OR HOME NETWORK ARCHITECTURE]

[keep this security oriented, Forget sections & subsections style.]

3.1 Home network architecture and IETF

Home network is a computer network located at person's home. It consists of devices and their connections, either wired or wireless.

This thesis avoids using term *homenet* when meaning network at home, because it is reserved to Internet Engineering Task Force Working Group's (IETF WG) homenet. IETF is responsible for the most Internet technology standards and WG homenet was started in year 2011. Current drive in homenet management is towards IPv6 environment as it allows future addressing and routing needs. As old technology cannot be forgotten, home networks will be heterogeneous having both old and new technology, and their interoperatibility is important in planning future home networks. Segmenting home in multiple subnets will also belong to homenets and will include areas for home members, guests, and management. Lastly, end-to-end access, is in their agenda. End-to-end access, i.e., restriction-free access was the key element for the Internet's success and it enabled many new applications in the past, but has then had difficulties because of firewalls and NATs.

Securing home network and its router's configuration can be done for example first limiting access to their administrative ports with static or dynamic access control lists (ACL) in routers. To get through administrative ports, i.e., to login and make configuration changes, there exists either AAA or local authentication. Authorized agents can then make changes, either direct in the device or through some management protocol such as SNMP or NETCONF[source needed?]. SNMP has been in use for over 30 years and is well supported in routers. Yet there are multiple version for this protocol. While earlier versions (v1, v2) did not provide any encryption of messages, version 3 knows for example about public keys and is secure enough when used correctly.

Customer Premises Equipments (CPE) such as ADSL broadband routers or set-top boxes, connect customer's network to operator's network. Management of CPEs on the border of home network and operator has existing protocols. For example, TR-069 standard[43] for CPEs has been used to implement self-configuration architecture in home networks[33].

RFC7368 about IPv6 Home Networking Architecture Principles from Arkko[3] defines the borders of the home network and states that internal borders in home network should possibly be automatically discovered. Limiting those borders to specific interface type would make it difficult to connect different realms locally. The same document continues stating that while home network should self-configure and self-organize itself as far as possible, self-configuring unintended devices should be avoided and let home network user decide whether device becomes trusted. So, these statements reveal us that home network environment still needs external configuration even with the proposed automation aids.

Homenet WG proposes the use of Public Key Infrastructure (PKI) at home. The public key cryptography is processor intensive and its asymmetric keys are usually used just in the beginning of communication. There they can be used to securely negotiate symmetric keys which allow faster cryptography processing.

To use PKI, bootstrapping protocols are first needed for trust anchoring and AuthN. Despite the etymology of name bootstrapping, "Lift oneself by his own bootstraps", bootstrapping usually needs some input from outside. For that, draft from Behringer[6] proposes, that one device is first chosen as a trust anchor and trust is built upon that anchor. This anchored device then becomes homenet's Certificate Authority (CA) service. In the end, the rest of the homenet devices need to apply for certificates from that CA to get under same trust circle. Key creation, key exchange and their usage is explained in similar draft from Pritikin[32]. There is also discussion about using manufacturer provided device certificates as trust anchor.

This model could also be expanded to a full ticket enabled Kerberos style network, where time-limited tickets (tokens) for both authentication and authorization exist for different services. Trusted Third Party authentication center would be setup with the help of MNO. One service would then authenticate an entity, here smartphone, and give it a time-limited ticket as a proof that the entity has been authenticated. When the entity wants to connect to the service, it asks from the central server again for ticket, but this time for the service by presenting the authentication ticket. In return it receives a service ticket which it can present to the wanted service. If

EAP-SIM was applied in such environment, it would be used only once, namely in the bootstrapping phase to setup the CA trust anchor.

Homenet configuration itself is mostly excluded from this work. For example, it is desirable, that changes in homenet are done only through local controller, not at local device because of synchronization issues, even if synchronizing algorithms such as Trickle[19] were used in homenet for configuration propagation. As another example, configuration also includes power level settings of devices to save electricity based on usage profile. For example at nights or when there is nobody home, some devices do not need to be working at their maximum capacity.

Instead, we study interfaces of AA. Main points here are an existing infrastructure (phones, internet access, Wi-Fi access points), a strong authentication (two-factor), and authentication methods (EAP-SIM, EAP-AKA, EAP-AKA').

3.2 Centralization trends in management

Traditionally, management of network devices has been done individually using each device's console or web-access. As the number of devices has increased, it would have been reasonable to rationalize the process by utilizing a central management, not least to prevent human errors for repetitive tasks. Yet, at home, network devices often are too heterogeneous, bought at different times from different vendors and so incompatible with each other to fully benefit from centralization.

To help moving the management to a more centralized model, the home network will see smartphone a central managing local controller. Usually, home users already have a phone, which can be considered as 'smart'. Most smartphones have Wi-Fi capabilities and writing programs for them is possible even with only little knowledge. When we choose a smartphone to be the management point, the other benefits are numerous: a management software can be delivered and updated from cloud to diverse smartphone types, existing user base having smartphones is orders of magnitude more than in any single organization, and as the most important fact, the trust anchor can be set to the smartphone.

The users are already located centrally in operators' user databases in Home Location Registry Authentication Center (HLR-AuC). To be able to achieve the management paradigm change to a central configured one, we still need to bridge homenet to that model with a trusted local controller and resolve the work-flow of change management.

4. DESIGN OF HOME NETWORK TRUST ANCHOR AND SEPARATION OF CHANGE MANAGEMENT

This chapter describes, how the change management can flow after the home network has received a trust anchor in the smartphone. Before fully explaining our chosen method, we introduce some alternative approaches for a trust anchor, which is part of bootstrapping. Bootstrapping is needed because although the smartphone and MNO already trust each other, the trust between the smartphone and AP, and thus the management network at home, is non-existing in the beginning.

On its simplest, the smartphone connects with a Wi-Fi link to an AP in the home network and authenticates with SIM-card. That brings a trust relationship between smartphone (local controller) and home network (managed devices) so that the configuration changes can happen as long as the smartphone is present.

4.1 Alternative methods for introducing trust anchor into the homenet

Trust information, may it then be a secret or some other evidence, can be delivered to a trust device via physical transport channel separate from the actual communicating channel. Traditional way to do that is with a password inside a sealed envelope or a one-time password list that for example online banks use today. The secret can also be sent as an SMS.

Trust can also be requested with the help of device's unique properties. Recently there have appeared devices on the market that have vendor certificates inside them[cite], which brings public key infrastructure as one possible alternative to learn trusted identity. The device proves its identity by presenting a certificate, which has been issued by a trusted vendor. Private keys are inside the device's trusted hardware store. Vendor-trust is needed for checking the issued certificates and so the trust verification is merely transferred from individual devices to trust

verification of the vendor. Root CAs are trust anchors also and can in the same way be read from the device's read-only store. CPE could use vendor issued certificate for AuthN of some earlier unknown device. If keys are stored in SIM as they here are, an external operator support is needed.

[Picture]

Other techniques to use SIM's unique properties besides EAP-SIM are for example Bluetooth SIM Access Profile(Bluetooth SAP), direct connection through PC/SC (PersonalComputer/Smart Card), CallerID service from phone network, and Mobile signature service such as "Mobiilivarmenne" in Finland.

Bluetooth SIM and PC/SC would need patching of smartphone's software to work. On the other hand, the smartphone would any way need to download a controlling application in the beginning for advanced use, so these techniques could be studied further in another work.

Caller ID as an authentication method uses GSM network's controlling channels. When a phone makes a call, the receiving end gets to know callers phone number (IMSI) before it answers the call. That information is called Caller ID and it has been in use successfully for some door locking implementations. It does not cost anything for caller or responder, because after receiving the CallerID information, responder can hang up upcoming call and no call expenses are created. It can also be made safe at least in Finland by limiting which tele operators are allowed to connect.

European Telecommunications Standards Institute (ETSI) defined a standard for mobile signature services (MSS) in ETSI TS 102 204. MNOs in Finland have implemented this as a service called "Mobiilivarmenne". For example, MNO Sonera's brand for it is "Sonera ID" while MNO Elisa calls it "Elisa Mobiilivarmenne".

When AuthN and AuthZ comes from outside, one possibility is to use a federated Mobile AuthN Service, which then is connected to MSSP(Mobile Signature Service Provider) with ETSI-204. Benefits for ETSI-204 federation is that no single home device must implement it at home, but also MNO sees service as just one client. Without federation, mobile AuthN services would need to be multiplied with number of the separate homenets, which need authentication service.

Project Moonshot, if worked and used together with MSSP, may offer SIM-based SSH-access to Authenticator. Modifications are then needed both in SSH server and

client. Additionally EAP must be used through tunneling, for example as an inner protocol of EAP-TTLS.[24]

At this point question might rise, why these external service providers are needed. Is it not easier and simpler to just send an SMS with password code to the smartphone, when access confirmation is needed? Mobile SIM provides two-way AuthN part as discussed earlier.l Without need for strong AuthN, that model would indeed be simpler, but using SIM also solves initial key distribution problem. Additionally, mutual AuthN problem would still need to be solved: Who sent that password?

All this time it is assumed, that hardware does not lie. In case the hardware has been tampered, we could not trust it and its claims. For example, there have been attacks against SIM to reveal its private key after SIM have been copied. To verify, that a device has not been tampered, a method called attestation can be used. A device which has attestation capability such as hardware certificates or Trusted Platform Module (TPM) technology can function as a trust anchor. Such a device could be sent direct to customer with pre-configured secrets and methods to take a place as a trust anchor. That leads us again to the key distribution problem.

There is also fraudulent Authenticator problem: the Authenticator may present some information to the Authentication server and other to the EAP-peer. Mitigation for that is, that EAP-peer includes some characteristics of the Authenticator inside its EAP-message, which then the Authentication server verifies (RFC6677)[13].

The phone brings trust to the homenet by completing full EAP-SIM AuthN through the local Authenticator. SIM's identity is verified by HLR AuC at the phone operator's end. The verification leaves a trail on the local Authenticator and opens a trust channel for a limited period of time for changes from the phone. [This was the most important paragraph of whole work. Thanks for reading it.]

Requirement for homenet can be as small as having WPA2 Enterprise capable AP. Almost any AP will do, but as an exception, cable modem Bewan, which has been distributed to many homes from the cable modem operator Elisa, was found to have only WPA2-PSK mode. Additionally, managing user's SIM-card has to be registered as an admin user in homenet configuration, i.e. IMSI must belong to the admin group. In this implementation, no extra application is needed in smartphone for primitive trust, but later for more serious use some application is needed. For added functionality, for example for logging admins out, OpenWRT based software can be used, although those functions have not yet been implemented. Disconnection issues are explained in Section 5.3.

4.2 Flow of design [TBD already above, FIX THIS section]

To demonstrate how the model works, we present the case of adding a new admin user.

Let's first suppose, for case of simplicity, that the homenet has been already configured(bootstrapped) and it is functioning properly. The home configuration model has been copied[inserted, etc] to the cloud. When changes are made to the cloud model through authorized cloud administrator users (operators), those changes are later also committed in to the production in homenet. There is no magic here, plain configuration change, just this time externally initiated.

Now, let's think what happens, when the cloud operator (or owner of homenet) tries to modify attributes, which give access to a new actor, such as a new operator, who would want to have access to separate segments of home network. First we need to have that segment separation change approved and after that we want to allow the newcomer account to have access to that segment and only to that. For the first part, which is normal operation, approving would perhaps yet not be necessary, but for the second part we need some checking unless our trust to cloud operator is ultimate. [FOR approval needs, discuss this with the team.]

When CPE of homenet is about to input configuration changes which would change balance of authors or roles (if role-based authorization in use), it needs to check if that is permitted. Permission would need to be asked from trusted point, here mobile SIM but instead of that the CPE checks from its state database, whether mobile SIM has been given access to management network. CPE wants to verify, if the changes are authorized. They are, if currently smartphone user is logged in management network, i.e. management is allowed.

Alternative method is that the changes could be marked some way, so that they need approving and then there could be a specific change-approval message, which must be sent through management network, perhaps including digest of change message as a verification.

Because smartphone is not actively listening the CPE, it cannot input those request. There are three planned ways to distribute changes.

1. Changes are delivered normally from cloud to CPE (CPEs) without interaction from the smartphone. Such changes would not need AA at all or changes include credentials to login to targets.

2. Changes are delivered from cloud to CPE functioning as a central management station without interaction from the smartphone. Digest of what is going to happen would be sent to smartphone from BaaS over the air (OtA). Smartphone would authenticate in to management network (if not already there) and send through it the digest token it received from cloud as an approval message to central management station inside homenet, which then forwards configuration changes to other devices.

3. Changes are delivered from cloud to smartphone, which after authenticating into management net, forwards them through management net to each and all devices.

The smartphone may receive the authentication token with a message explaining what is going to happen in the change. As the CPE and the Authenticator may be separate devices, approving happens by sending the token from the smartphone to the CPE via the management network where the Authenticator gives access.

It must be noted, that the smartphone can already have an association to a non-management network with Wi-Fi. If that is the case, it first must disconnect from there and then connect (i.e. AA) to the correct management network. That implies disconnection from other services using Wi-Fi link, because smartphones currently have only one Wi-Fi radio available and routing prefers Wi-Fi as a default gateway, although possible 3G data link still may stay operational.

4.3 Chosen design and why (Rationale)

Network can be divided into separate segments. First, there is normal access network which provides connectivity. Second, there is network through which devices are managed, so each device need to have at least two connections: one for access and one for management. It is not defined, if those connections are physical or virtual (VLAN, Virtual LAN). An analogy to the real world would be public access corridors and doors for customers separate from privileged doors for service personnel.

Access to the network segments is checked in routers with access control lists (ACL), where decision is made based on current configuration or user's role. Once user has been authorized into management network, access stays open for him, at least for a (predefined) limited time.

So, instead of checking user's credentials each time data is received this model only checks, from where data is received. Data received from the management network

is granted for changes. It is arguable a lighter method than always fully AuthN and AuthZ but may suffice here, at first.

Example of a complex solution would be a traditional firewall and packet inspection in the interconnects, but very modern model is the de-perimeterization trend set originally by Open Group's Jericho Work Group in 2004, that won't leave trust verification to perimeters of network (firewalls and application proxies) but always handles traffic as coming from untrusted source.[28] One implementation of de-perimeterization is Google's BeyondCorp[42], where all traffic always travels through Access Control Engine and is suspected as being external, even when it originates from inside networks.

In production, some changes in local controller are propagated to homenet via management network without need for an extra authentication phase. The local controller does not interact there. An example of change is a modification in network segment, which does not change network topology of other domains. Those changes or alternatively changes that do need authorization should be enumerated, which ever would be smaller set.

In our model, only initial bootstrap needs the authentication with smartphone as well as change of admin roles and some dangerous combination of commands.

[sync. part to misc Section ?]

When home network needs secure binding to the smartphone, earlier mentioned trust is the first one needed. The trust is achieved by checking whether the smartphone can access home management network using only its trusted SIM-card, which provides AuthN. AuthZ in turn is compared to existing roles of IMSI in the Authenticator.

[This has been explained in 802.1X Section in the begin. TBD]

Technically we use in Wi-Fi connection IEEE 802.11i (also known as WPA2), which includes 802.1X as port based access protocol. 802.11i defines there authentication, authorization, and cryptography key agreement. It uses EAP for selecting authentication mechanism, after Authenticator requests smartphone to identify itself as in Figure xxx is shown Messages are carried over 802.1X or RADIUS depending on transport medium as of Figure 2.1.

When AP forwards authentication request to next RADIUS server, it can ask or receive, beside AuthN and AuthZ, other service parameters, such as provisioning. That would allow the smartphone to connect to specific management network access either via CLI or SNMP or similar[26, p.4]. RADIUS can carry extra attributes in

its ACCESS-ACCEPT message. In essence, AuthZ part itself can be thought as one type of service provisioning.

There exists RADIUS attribute types for directing user into specific VLAN. If those do not suffice, there is also special Vendor Specified Attributes (VSA). VSAs allow vendors to define up to 255 own attributes that can be used in provisioning in homogeneous environment.

That way (3rd party) Authentication server can decide which network segment the device would be put. In our case, admin users are put in to the management network. Yet, usually RADIUS ACCESS-ACCEPT message, which means AuthN and AuthZ were successful, puts the user in default network, i.e., just gives it basic access. As for other provisioning parameters, not all end devices support them.

In the first prototype it is enough to identify authorized smartphone's SIM. Smartphone holding the SIM is granted the access to the parts of the management network and is authenticated strong. User management is outsourced to MNO, which already has provided SIM cards to users. What remains, is the adding of the user's IMSI to the authorized users' list. That list can be located on diverse place, as can be seen in xxx

After authentication and authorization has succeeded, session key creation occurs (WPA2 session) between AP and the smartphone. The Authenticator has opened port to the smartphone for configuration changes. The Local RADIUS (if existing) has trails of successful authentication and knows which IMSI has successfully authenticated in the home net. It also knows mapping between IMSI and temporal IMSI for cases where the smartphone later would need re-authentication.

4.4 [Need for Security bootstrapping]

[removed, NOT YET trust anchor methods HERE!!!]

[Description of General Bootstrapping architecture (GBA) vs. yet another custom architecture. Maybe parts of architecture such as using SIM-auth (EAP-SIM) or CallerID, how they differ. What is needed? How GBA could be used here?]

In Behringers work-in-progress bootstrapping[6], AuthZ happens likewise first at cloud provider's end, but after checking device's Vendor certificates, cloud provider gives device a ticket of authorization like in Needham-Schröder or Kerberos implementations. Device presents that ticket to CPE which finally can decide, whether

it allows change. Instead, here the Authentication server can be external RADIUS server, but usually the final decision point lies at the Authenticator in CPE.

4.5 Access methods to Wi-Fi with only one SSID

[To be cleaned!]

Today, homenets usually consists of only one Service Set ID (SSID) Wi-Fi network though it is possible to define multiple SSIDs in an access point. Having multiple SSIDs enables us to dedicate one of them to management network. To enable EAP-SIM method, it is necessary to use WPA2-Enterprise mode and thus RADIUS server.

It was not found, how Authenticator could use the same network with both WPA2-PSK (or open access) and WPA2-Enterprise, so separate SSID for management network was technically needed. If Wi-Fi was limited to only have one SSID, then we would need another way to separate access requests to management net. Access to Wi-Fi can be separated by multiple realms (different username domains), different authentication methods, or user's role given by Authentication server. Management through Wi-Fi has then three options. Without RADIUS, access is open and the only checking comes from the used management protocol and its access control.

[2015/05/11 NEW! This must be told everywhere, devices still have their own access control! Or do they use RADIUS? Now RADIUS is used to carry on EAP auth to get into access network, why not use it also to get in device?]

With WPA2, PSK is used, but no EAP or RADIUS as backend. With EAP, RADIUS server is the one who returns correct values to get in management network in ACCESS-ACCEPT message as was explained in Section 4.3.

4.5.1 HS2.0 [If deleted, remember also from conclusion! TBD]

Wi-Fi Alliance has certification program (Passpoint) for Hotspot2.0 compatible devices. Hotspot 2.0 enables selection of network based on ownership, services and performance characteristics *before* Wi-Fi client has been associated to Hotspot 2.0 AP. The technology is built on IEEE 802.11u specification.

It is well known, that usability of Kiosk-mode Wi-Fi networks is burden, because user needs to go through web portal logins with username-password authentication procedure and those are different for every network. HS2.0 would help there.

In `http://www.ericsson.com/res/thecompany/docs/publications/ericsson_review/2012/er-seamless-wi-fi-roaming.pdf` goals are to smooth roaming between Wi-Fi and 3GPP/LTE networks and bring operator-grade to Wi-Fi by putting control in operators side. More than offloading traffic, plans are to bring other services also to Wi-Fi.

TO DO: check 802.11u features and what they add to 802.11-2007

- interworking with ext networks
- hs2.0 is extended 802.11u
- next generation Hotspot
- advertises external networks *before* association. no need to select Service Set ID (SSID)
- access network type, roaming consortium support and venue information
- some QoS mapping
- emergency services (not in HS2.0)

4.6 Scenarios for choosing the AuthZ location

AuthZ decision usually happens at home and the Authenticator gives anyway the final decision about access. If the decision is made on remote AuthN server, 3rd party, then that server needs to have either local data or access to cloud service's AuthZ data. Further it seems inevitable, that just like the home network model having AuthZ data of eligible IMSI accounts is in the cloud, then also delegating AuthZ to cloud would simplify home network functions. Instead of putting logic on CPE for AuthZ, CPE could just trust the 3rd party service's AuthZ message, which in case of RADIUS is either *ACCESS-ACCEPT* or *ACCESS-REJECT*.

Here are 5 scenarios presented for possible location points of AuthN and AuthZ. Authenticator is the entity which gives the final decision about access regardless of location of AA. In most cases it is located in the local AP, but it can also be external, like in scenario V in table 4.1, where locations for Authenticator (AA), AuthN, and AuthZ are marked as (I) for internal or (E) for external.

[1] BaaS provides

Table 4.1 *Location of AA, AuthN and AuthZ in scenarios I-V*

scene.no:	AA	AuthN	AuthZ
I	I	E	E
II	I	E	I
III	E	E	E
IV	I	E	E[1]
V	-	-	-

The first AA-scenario is presented here thoroughly as an example. The goal is to make trusted configuration change. The steps are numbered in Figure 4.1. Configuration change is allowed, if CPE gets ACCEPT from MNO. MNO gets information of allowed users from Cloud (BaaS [def.])

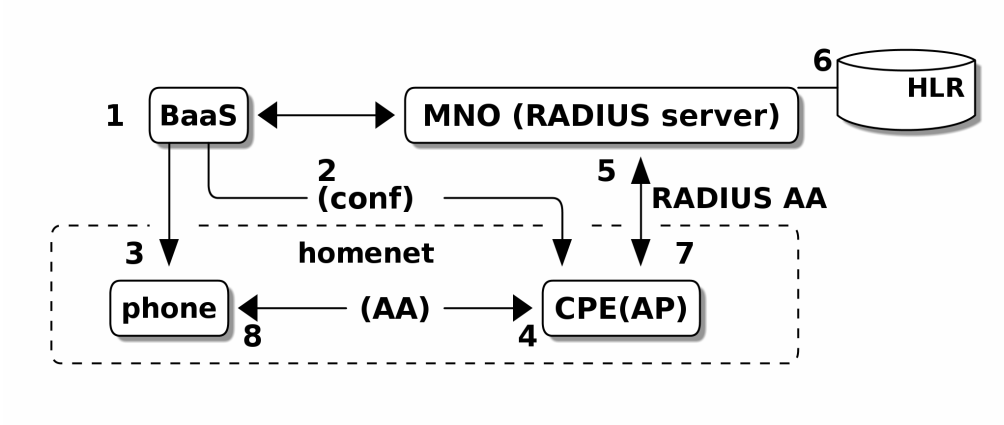

Figure 4.1 *Scenario I with 3 separate domains: BaaS, MNO and homenet*

[Maybe replace BaaS with CLOUD]

[alternative presentation of the flow, list]

1. The model has been changed in the BaaS (1).
2. BaaS send changes to CPE (2).
3. If changes are privileged, they need to be approved by phone user. Changes are sent also to the phone(3) and phone user must authenticate itself to the management network.
4. Phone user starts authentication process to management network using EAP-SIM and reveals its IMSI(4).
5. CPE (AP) forwards authentication to MNO's RADIUS server with RADIUS protocol (5).

6. MNO have RADIUS server running and it authenticates IMSI user with its HLR-AuC (6). MNO also asks from BaaS, whether IMSI user has admin-role (AuthZ).

MNO returns in RADIUS message either *ACCESS-ACCEPT*, if user is both known AND has admin role or *ACCESS-REJECT* (7).

1. CPE receives this ACCEPT or REJECT. If there were other RADIUSes between CPE and MNO, they would have acted as proxy RADIUS servers.
2. IF ACCEPTed, then the smartphone is both authenticated and authorized (8) and can send configuration change message to CPE, which recognizes it coming from authentication network.

[alternative presentation of flow, textual paragraph. Which one is better?]

The model has been changed in the BaaS (1). BaaS send changes to CPE (2). If changes are privileged, they need to be approved by phone user. Changes are sent also to the phone(3) and phone user must authenticate itself to the management network. Phone user starts authentication process to management network using EAP-SIM and reveals its IMSI(4). CPE (AP) forwards authentication to MNO's RADIUS server with RADIUS protocol (5). MNO have RADIUS server running and it authenticates the IMSI user at its HLR-AuC (6). MNO also asks from BaaS, whether IMSI user has admin-role (AuthZ). MNO returns in RADIUS message either *ACCESS-ACCEPT*, if user is both known AND has admin role or *ACCESS-REJECT* (7). CPE receives this ACCEPT or REJECT. If there were other RADIUSes between CPE and MNO, they would have acted as proxy RADIUS servers. IF ACCEPTed, then the smartphone is both authenticated and authorized (8) and can send configuration change message to CPE, which recognizes it coming from authentication network.

While changes already have been sent direct to the CPE and only let it wait for approval, then when CPE receives ACCESS-ACCEPT, it could proceed on propagating those changes. Otherwise, after certain timeout, CPE must stop waiting for phone's approval and drop changes.

This simplification has pitfalls. If the smartphone stays in management network continuously, how upcoming changes are separated? The smartphone should either be dropped out from management network right away after changes or after predefined timeout period. If on the other hand, the smartphone must send changes

itself, then it would be possible that access in the management network has short period of time, when phone holds that status or acceptance token. For example for 10 minutes connection would be open for changes. Then changes would not go directly to CPE but instead to , but they would include some token to phone, which is needed for approval message.

In second scenario (Figure 4.2), AuthN is asked from MNO but AuthZ is checked from local database. Local data comes from data model i.e. from configuration data and will be saved in CPE, or some other place within homenet.

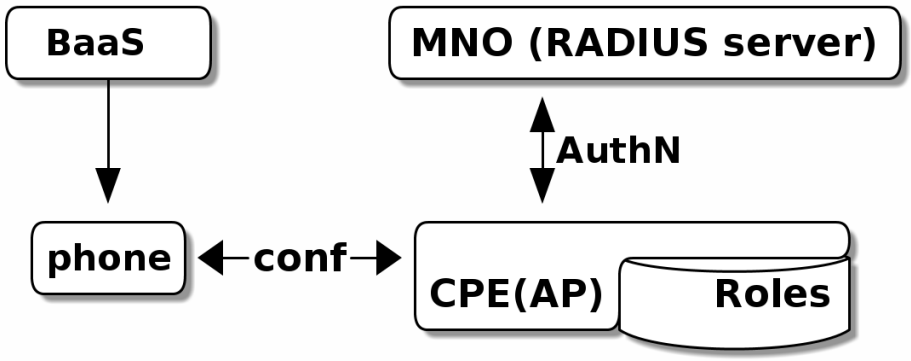

Figure 4.2 *Scenario II with AuthZ in homenet*

Similar to first scenario is scenario III (Figure 4.3), but this time there is SP between CPE and MNO, so AA is fully outsourced: local AP communicates with RADIUS protocol to the external Authentication server. That in turn gets AuthN from MNO via its hlr-auc-gateway and AuthZ from BaaS. Locally there is a cache for roles in case of network disconnectivity.

Here benefit is, that 3rd party Authentication server may have direct contracts to many alternative MNOs, so user does not need to find and choose them. As a bonus, MNOs already delegate requests to right operator, if they happen to get AuthN request which does not belong to them. This is similar to federated service.

Allowed users are verified from BaaS's registries and specific IMSI is authenticated from MNO. It may need some preparation, if SIM identities are temporary i.e. TMSI is used. Still, IMSI is carried out at first message of full authentication. Later, the server would need to have mapping between IMSI and TMSI, but because only full-authentication is used, there should be no problem.

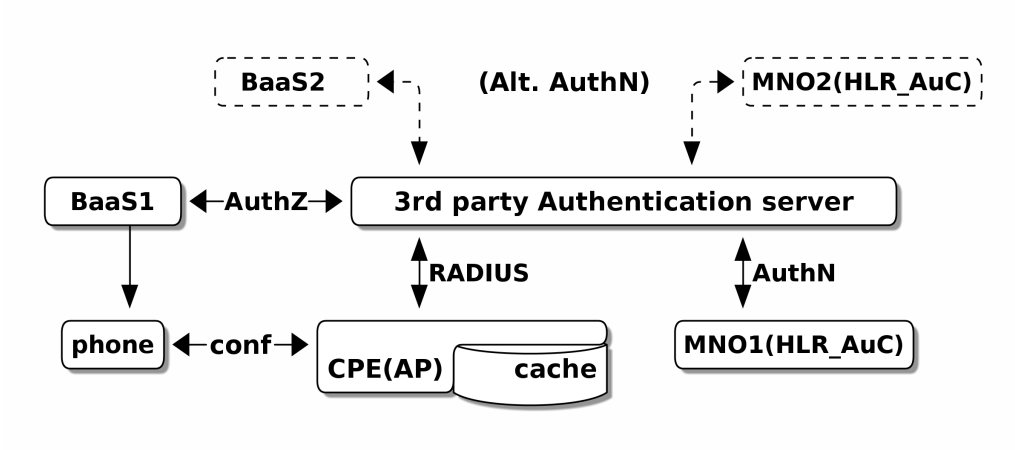

Figure 4.3 *Scenario III with outsourced AA*

Scenario IV (Figure 4.4) is similar to scenario I, but now AuthZ is checked by CPE instead of MNO from BaaS. If there are no connection to cloud, fall-back is to work just like II. So also this scenario needs local store for admin IMSIs.

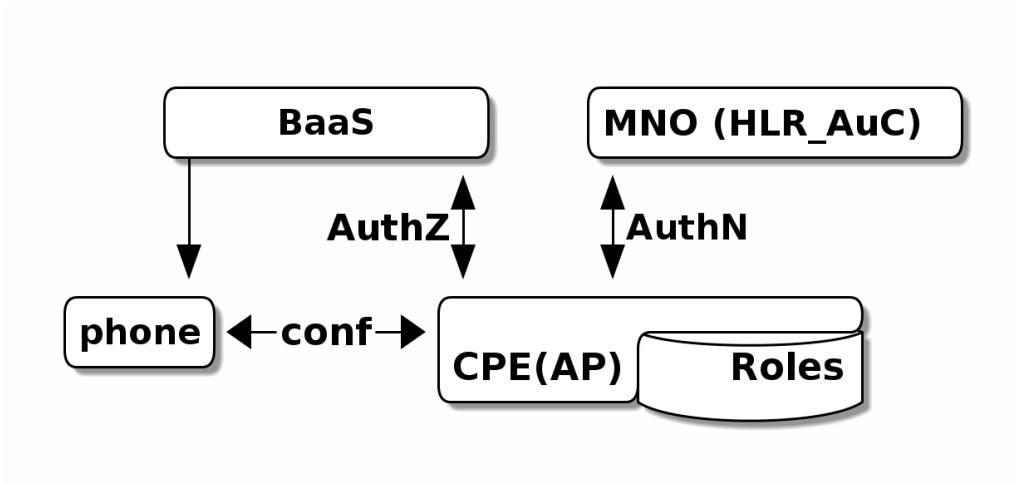

Figure 4.4 *Scenario IV, AuthZ from BaaS, AuthN from homenet*

In the last scenario (no figure), nothing has yet been configured. The bootstrapping is not done yet. The scenario can be any of I-IV, but no trust nor roles are present in CPE.

4.7 Ways to modify RADIUS messages [perhaps to security integrity chapter?]

RADIUS messages are not protected from eavesdropping, but they have integrity fields to notice if tampering has been done. Integrity field is called a Message Authenticator. Notice the use of the term *Authenticator* in different context here,

not meaning 802.1X's Authenticator. When using RADIUS to AuthN and AuthZ, Requests can only belong to ACCESS-REQUEST messages while Responses can be any of ACCESS-ACCEPT, ACCESS-REJECT, or ACCESS-CHALLENGE message. The Message Authenticator field is sent as last Attribute Value Pair (AVP) of each RADIUS message and it can belong to either Request or Response.[14, p.20].

The Request Authenticator is 16 octet long, random number in ACCESS-REQUEST message but the Response Authenticator for it is achieved by one-way MD5 digestion function. The digest is taken from concatenation of Code, ID, Length, corresponding RequestAuth, Attributes, and a Secret and can look like $3fef65608 \ldots 2a79$.

```
Response Authenticator =
    MD5(Code |ID |Length |Request Authenticator |Attributes |Secret)
```

The Secret is the shared secret which has been configured between RADIUS servers, and it protects some parts of traffic. Different RADIUS clients may have different secrets and RADIUS server must separate them by client's IP address to manage proxied RADIUS requests[14]. If the user password was to be transmitted on wire, it would be run through exclusive OR function (XOR) together with MD5 digested Secret and Request Authenticator.

```
User-Password = XOR(password, MD5(Secret | Request Authenticator))
```

Our model would greatly benefit from modification of RADIUS messages in proxying RADIUS, if that is possible as was mentioned in Section 2.2(RADIUS). The modification is needed when proxying RADIUS combines AuthN message from MNO to AuthZ decision received from elsewhere.

RFC6929[8] reminds, that even when the proxies do not understand all AVPs inside RADIUS message, they must deliver those values and that allows us to use larger set of AVPs than is in any (proxying) RADIUS server's vocabulary. By adding AVPs inside the authorization packet, we achieve extra information about validity of the access request. RFC2865[34] says, that the forwarding RADIUS proxy may alter the packet as it passes it, but because an alteration would invalidate the packet's signature, the proxy has to re-sign the packet.

So at least Proxying RADIUS can insert something, but that is not enough. If a malicious actor imitates RADIUS Proxy (i.e. Man in the middle, MiTM) and tries to inject untruthful messages, Message Authenticators might help in detecting that. Unfortunately MD5 hashes were first time broken by brute force already 20 years

ago and today they can only be used as data error detection[40, p.2]. MD5 can not be thought as computationally secure, because duplicate hashes are easy to compute today[45].

4.8 Similarities with Lock-and-Key method

The method is very similar to the concept used on routers to dynamically enable access to certain parts of network by first letting the user to log in to the router.

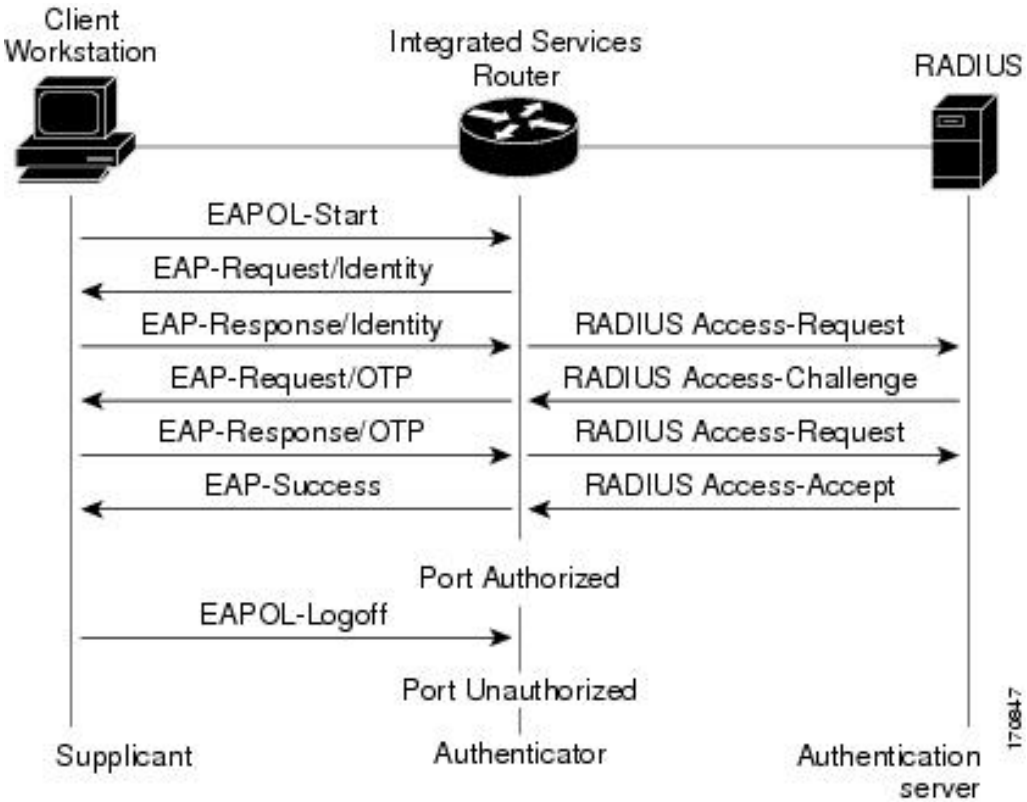

Figure 4.5 *802.1x access control with EAP [TBD deleted?]. Source Cisco.*

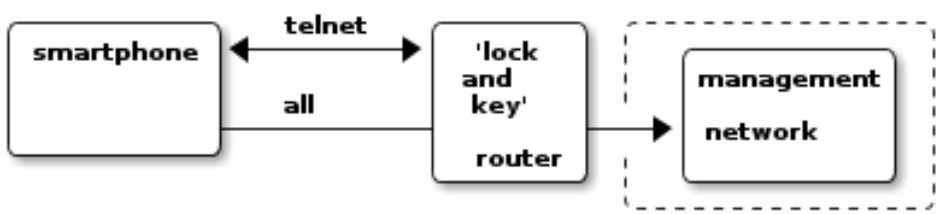

Figure 4.6 *Cisco's view of Lock-and-key access*

Device provider Cisco calls this "Lock-and-Key"[30, p.117] access and uses dynamic access list to implement it. Difference here is that 802.1X protects access to the

network in Layer 2 while Lock-and-Key needs to have a functional Layer 3 to conduct authentication phase. Figure 4.5 reminds us again how 802.1X works with respects to Lock-and-key in Figure 4.6.

Smartphone has only limited access to the network before AA has completed, while in the Lock-and-key the other parts of network are already open and successful login to the router opens access to even more segments through it. In other words, Lock-and-Key protects IP-access in layer-3 , while 802.1x protects layer-2. Both methods can have RADIUS as an Authentication server. When RADIUS is not available, for example because internet is down, there almost always exist as a failover a local password method in the configurable router.

[This belongs to multiple SSID section]

If Lock-and-key method was used instead of EAP-SIM RADIUS, then separate management LAN would not be needed. Roles were given at Authentication server or designated router after the smartphone has done login to it via normal access network.

This thesis suggest a mix of these methods: EAP-SIM 802.1x WPA2 for getting in the management segment ...

1. Smartphone connects a Router via wireless AP, and needs to login
2. Smartphone uses telnet (or ssh) to login to the ROUTER. (but with which credentials?)
3. ROUTER(as RADIUS client) checks AA from Authentication Server(or proxy)
4. AA-server answers based on earlier SIM-authentication that this request is correct

4.9 Summary of the chosen solution]

[wrap up of solution]

The chosen solution to benefit from SIM is via EAP-profiles, as EAP is well known when using WPA2-Enterprise protection in Wi-Fi.

Design is [move from above]... and it is a variation of lock-and-key design.

Above it was mentioned, that the local controller delivers changes to each device. On this work, it is first assumed that the local controller (smart phone) only *approves* changes, and delivers them to *one, central CPE*, which handles distribution of changes to other CPEs. The distribution is not further described. Later, the Authenticator is both the AP and RADIUS client (in scenarios I-V), which receives RADIUS messages from Authentication server, even when there would be a separate local RADIUS server running as a proxy. Lastly, a variation of the design is, that not every change needs to go through the local controller and so the process does not always need interaction from the user. For example, if 3rd party has been given a right to switch on and off its sensor network, it would not be necessary for the home owner to accept those changes every time they occur.

Critical changes are those, where network topology changes so much, that different players would get access outside their earlier domains. Different players include external service providers, users at home, visitors, and also home net owner. Examples of the previous cases can first be seen on the division of homenet to guest and private network and extensions for homeworkers instead of office.

5. IMPLEMENTED SOLUTION

To prove that the proposed model works, empirical tests have been done. First it is shown how EAP-SIM authentication works. Then a use case for adding an admin user is reported. Changes are in the end done for example with NETCONF from the management network.

5.1 EAP-SIM authentication test bed

Used physical devices were two smartphones, an AP and a laptop. The smartphones were Nokia E70-1 and Nokia E90, both capable of EAP-SIM. The AP was running OpenWRT firmware. Laptop's software were WPA2-Supplicant for Wi-Fi 802.1X access, hostapd for wired connected RADIUS server and hlr_auc_gw for MNO's HLR-AuC. Laptop's role was therefore physically split-brain: It asked from itself for AA. Figure 5.1 shows how EAP-SIM AuthN messages (dashed and solid arrowed lines) flow when using simulated WPA2-Supplicant and HLR-AuC as simulation environment.

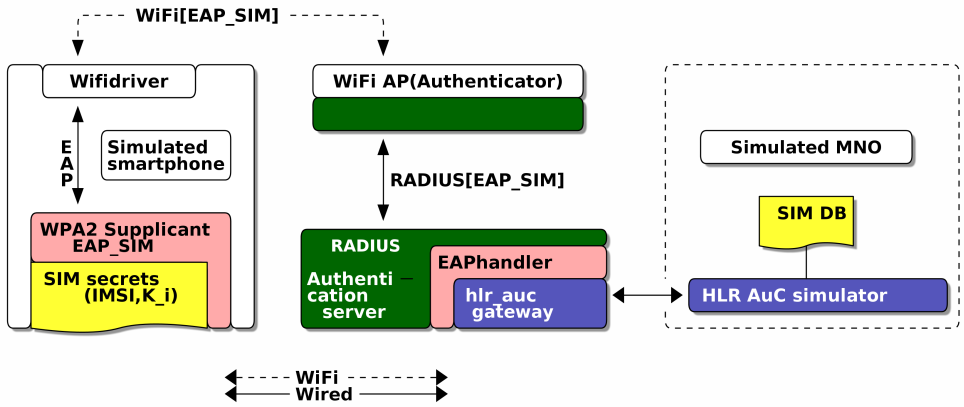

Figure 5.1 *EAP-SIM AuthN messaging in simulation testbed*

Jouni Malinen's software package *HostAP*[23] can be thought as a reference implementation providing all necessary components: WPA2-Supplicant, Wireless Access point (AP), HLR-gateway (for GSM networks) and EAP-endpoint with or without

RADIUS-server. HSS would replace HLR in 3G/UMTS networks. The version used in the tests was 2.2, while version 2.4 was published on March 2015.

For a more realistic test, OpenWRT AP is used instead of *hostapd*'s AP and *hostapd* provides only RADIUS server. OpenWRT AP works as a RADIUS client connecting to RADIUS server. It will not try to open EAP-messages or need to know about them; it just encapsulates them into RADIUS packet.

The algorithm used in the demo was internal GSM-Milenage, which handles beside EAP-AKA also EAP-SIM. Milenage is a reference implementation and as such suitable for operators, who do not want to invent their own security algorithms. OPc and Seq numbers, which are needed in when using EAP-AKA, were not used.

5.2 Detailed description of test runs

Test runs were made with diverse clients. Nokia E70-1 with Symbian 60 Series OS (2006) had a non-registered SIM card. Despite that it took part in generating primary EAP traffic. Examples in appendix A.5 [TBD]

First tests did not go as planned. There was no indication of SIM method present in captures, the only indication of security was message "Open System" in application logs, which means that no pre-shared key is used. Nokia E90, with a registered SIM had better results. Traces are in folder `gitdocs/di/testit/` files `eap3.pcapng`, `e90.sim.auth.pcapng` and `eap-1.pcapng` [TBD]

After some modifications, runs got to the authentication phase. Naturally, challenge-responses did not work because SIM secrets were not known. Nevertheless, both card succeeded to the point, where MNO's message would be verified with the SIM card.

Unregistered phone could not use SIM card while registered phone verifies and notices, that operator is not right, and therefor ends conversation as should be regarding protocol-document [EAP-SIM].

At this point, physical phones were put aside and simulated SIM-card was used. After WPA2-Supplicant run on laptop with simulated SIM-card access with SIM/USIM protocols, respective EAP-SIM, logging from hostapd software claimed that "Hostapd will send SIM/AKA authentication queries over a UNIX domain socket to an external `hlr_auc_gw` program." Appendix A.4 shows that traffic.

Tests were run with a shell program (Appendix A.1), which started the needed programs. It also recorded the used configurations, logs, and traffic captures for later analysis.

[Analyze configuration script here]

scan_ssid: When set to 1, this will add the SSID to the probe requests, in case you're connecting to a hidden network or an AP with multiple SSIDs.

5.3 Disconnecting the local controller and offline changes

[Limiting time and forced logout, for how long access provided to management operations, or use fast-auth on following accesses TBD]

After the phone has been successfully connected to the management network, changes coming from the phone can reach routers. There should be a way to close the session after the changes has been applied. Originally it was thought, that the session would stay open only for a limited time, after which the phone would be forced to logout or thrown away from the management network and that idea should be kept in mind when the final implementation is made.

Later it was learned, that terminating a session is not included in the original RADIUS protocol. The root cause is, that messages originating from the RADIUS server are not defined in the RADIUS protocol and so AP as RADIUS client cannot receive RADIUS server initiated disconnection messages. Additional extensions such as Disconnect and Change-of-Authorization (CoA) packets, also known as RADIUS Dynamic Authorization or RADIUS Disconnection Message(DM), have later been brought in[7] to the protocol by diverse vendors, but they may not all be implemented on every device. Disconnect-Request is sent to UDP port 3799, so Authenticator should listen also that in addition to RADIUS UDP port 1812. As a side note, Diameter protocol would provide server initiated messaging.

[Back in track: this can be left here]

- forced logout, like in captive portals, where RADIUS is not used.
- no straightforward solution exists within RADIUS portal [-> reference to No Internet connectivity link is

Offline changes include cases where the smartphone is not available or when internet connection is down. Full authentication uses IMSI, which is the identity of the

phone's SIM. Lighter, fast re-authentication would use temporal identity TMSI, which can change each time the AuthN request had been sent. Mapping of TMSI is cached on the Authenticator and the round-trip to and handling at HLR is so eliminated. Re-auth also works when there is no internet connectivity, but Full authentication does not work, because it needs MNO.

To defeat internet connection problem, a simple solution would be sending a one-time password to a predefined phone via an SMS, but what entity would then check that and who would be authorized to send that message? Authenticating server, which has no internet connection should have a way to check that one-time password received via SMS is correct.

Solution for this could be a co-existing WWW-based authentication, that is, a web-page where credentials could be entered. Software could run in AP. As AP is programmable with luci, it could run direct a WWW-access portal. Alternatively, existing programs such as *Chillisoft* or *NoCatAuth* would work as WWW-portals. For that to success, the WWW-portal would need also an open access without 802.1X port based access control.

5.4 Network traces (EAP, SIM, AUTH traffic analysis)

Wireless capture between WPA2-Supplicant and AP was made on WPA2-Supplicant's end-point, before it left the wireless card. Capture was not made in monitoring mode, so not all 802.11 details in data packets were captured[44]. Because the focus was not in the radio channel but instead in the EAP messaging, that was not problem.

[Captured wireshark sessions give insight here. Analyze them. Packet capture of successful SIM-authentication with corresponding parts of logs at WPA2-Supplicant, RADIUS server and packet captures 802.1X, RADIUS and HLR.]

- flow of messages, timing, size, attributes

Even when authentication conversation would not complete fully, Authenticator still receives identification claim from the smartphone. Yet, as there is no AuthN, no proof of identity exists in that case.

IMSI is sent first time already on the second EAP message from WPA2-Supplicant to AP (see Figure 2.3, message 2.) Same in tests made 150123-155714, source:

testit/demot/ap-s150123-155714/ Capture is from the the smartphone client, when it has received the first EAP packet from AP.

```
Frame 129: 15:57:17.983047
    Type: 802.1X Authentication (0x888e)
    Version: 802.1X-2004 (2)
    Type: EAP Packet (0)
    Length: 5
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 50
        Length: 5
        Type: Identity (1)
        Identity:
Frame 130: 15:57:17.983223
    Type: 802.1X Authentication (0x888e)
    Version: 802.1X-2001 (1)
    Type: EAP Packet (0)
    Length: 21
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 50
        Length: 21
        Type: Identity (1)
        Identity: 1232010000000000
#+CAPTION: EAP client's response to identity request
```

We note here, that AP uses version 802.1X-2004 while WPA2-Supplicant responses with version 802.1X-2001. Here it does not have any noticeable effect. The identity field's length is not shown here. It is not coded as a numerical but a string. That brings flexibility as the identity can include alphabets too. It also minimizes misunderstandings, if context gets lost.

EAP client's identity is transformed at Authenticator (Figure 2.1) from 802.1X's EAPOL format into RADIUS format and sent to RADIUS server:

```
Frame3: 15:57:17.988616
Radius Protocol
    Code: Access-Request (1)
```

```
Packet identifier: 0xa2 (162)
Length: 193
Authenticator: 055ff370b9e793c1e39d375aade8033c
Attribute Value Pairs
    AVP: l=18 t=User-Name(1): 1232010000000000
    AVP: l=7 t=NAS-Identifier(32): musta
    AVP: l=27 t=Called-Station-Id(30): 66-66-B3-8A-68-B3:simtest
    AVP: l=6 t=NAS-Port-Type(61): Wireless-802.11(19)
    AVP: l=6 t=NAS-Port(5): 1
    AVP: l=19 t=Calling-Station-Id(31): 5C-51-4F-E7-FA-F4
    AVP: l=24 t=Connect-Info(77): CONNECT 54Mbps 802.11g
    AVP: l=19 t=Acct-Session-Id(44): 5491885C-00000037
    AVP: l=6 t=Framed-MTU(12): 1400
    AVP: l=23 t=EAP-Message(79) Last Segment[1]
        EAP fragment
        Extensible Authentication Protocol
            Code: Response (2)
            Id: 50
            Length: 21
            Type: Identity (1)
            Identity: 1232010000000000
    AVP: l=18 t=Message-Authenticator(80): 04ea7e507d72bdb1acf515ef19ac9527
```

Interesting part is the EAP fragment, having Identity="1232010000000000", but also RADIUS message itself, where User-Name field has been set also to "1232010000000000". Identity is filled in WPA2-Supplicant both in identity and credential section so which one is the correct one, or are they both needed? Maybe this has something to do with identity values above, or then AP just has followed conventions on converting EAP into RADIUS message and put identity field into User-Name Attribute Value Pair (AVP). The last RADIUS (AVP) is Message-Authenticator, which presents limited safety against message corruption. Limited, because it uses MD5-hashing which is not safe against malicious use anymore.

[Here conversation]

[see. /home/itapuro/gitdocs/di/testit/demot/ap-s150123-155714]

6. ANALYSIS, RESULTS AND DISCUSSION

6.1 Deployment difficulty

To deploy the system, modifications must be done to AP and client. Additionally, contract must be made with the MNO service provider producing AuthN [while AuthZ is already taken care of with the cloud service contract.] [TBD, leave cloud out] For AP, modifications are minimal. Needed settings are WPA2 mode to WPA2-enterprise, IP-address of RADIUS server providing AA, and corresponding shared secret. For client, Wi-Fi profile must be added: used management SSID, protection mode 802.1X (or WPA2-Enterprise), and AuthN method EAP-SIM. Smartphone can have different profiles, also with under same SSID, but then user need to choose manually, which profile he wanted to use.

6.2 Estimating time to authenticate EAP-SIM [TO BE REMOVED]

Local tests, with software back ends need less than 20ms for one EAP-RADIUS message exchange between peers. There will be added time needed to scan Wi-Fi network for correct AP and SIM card's computing time. [Take reference on network authentication part on earlier tests. Timeout was 3 seconds for that part.] [Some Figures for authentication times can give comparison to eduroam or LANGATONWPA2 network through some RADIUS proxies in between home organization's RADIUS service.]

6.3 Costs for end-user

While no service yet exists from MNOs, we estimate their costs based on Mobiilivarmenne. Using Mobiilivarmenne is currently free for clients, if usage is personal, but costs for service providers are unknown. Regarding hardware, costs can mostly be eliminated, while users already have smartphones and for infrastructure, existing hardware such as APs can be used.

Using SIM to local Wi-Fi AA adds value to the smartphone ecosystem. To further divide possible costs for EAP-SIM usage is difficult. EAP-SIM always needs MNO for first authentication, because only MNO and SIM-card manufacturer know what are SIM's K_i and the used A3/A8 algorithm variation for GSM/3GPP/LTE authentication.

It is difficult to see if any commercial provider would implement SIM-key sharing so, that secret part were divided to a part that implements AuthN for own operator and to a part, that is free to use by some other operator. Instead, the same functionality can be achieved with Dual-SIM phones, which allow inserting two SIM cards from different operators in to the phone. By using menu option in phone, or even a specific prefix code before call, alternate SIM card can be chosen without booting the phone. Dual-SIM thus allows change of ID and IMSI without removing SIM card.

There exists also private GSM networks. An interesting use case of them has been Chaos Computer Club's international CCC-camps[12], where organizers provide private GSM network for attendees of conference by distributing them separate SIM cards for 2 euros. Even, when GSM network used 1.8GHz radio channel under an experimental spectrum licence, GSM encryption also could be used, because the SIM-card secrets were known to the organizing, private operator. On the other hand, empty GSM cards for testing can cost as much as 18 euros a piece (webshop-quote[36]).

6.4 Platform specific issues

For clients, there is no need for public key infrastructure (PKI) unless EAP-PEAP is used. Under PEAP there are either server certificate or additionally client certificate present. There are smartphones, that do not have EAP-SIM yet available. For example support for EAP-SIM (and -AKA) methods starts in Android only from version 4.x and in iOS from version 5.x.[18].

Generally, to support EAP-SIM with open source software in smartphone is *pcsc-lite* for accessing SIM card, *wpa_supplicant* for WPA2 client, and possible used connection manager (*connman* or *wicd*). This is in line, what was done in testing without *pcsc-lite*, because a file backend was used instead of a SIM card.

If OpenWRT platform is used for CPE, the memory size (32MB) restricts some use. WPA2 software included in basic OpenWRT installation is small, but that does not yet include RADIUS server part or EAP-SIM handling.

Software has other limitations. Freeradius2 is not yet included in OpenWRT. and if it were, it would also be based heavily on current Perl environment which itself needs a lot of space. Currently, as of 9.8.2015, Freeradius is running on version 3 and EAP-AKA is supported through a module from hostapd project. COMP128(versions 1, 2, and 3), which is implementation of A3/A8 algorithms, is supported[9], and so EAP-SIM is available. Yet, Freeradius can be used as Authentication Center (AuC). Diameter (freeDiameter) can be compiled in OpenWRT. That is good, because on 3GPP networks Diameter protocol has more support than RADIUS. If nothing else works, as a backup old-fashioned WWW-authentication portal can be used for offline authentication.

6.5 Security considerations

There can be multiple ways to attack the described methods of homenet management delegation. The following subsections divide them into confidentiality (privacy), integrity, and authenticity. Accessibility is also discussed.

6.5.1 Confidentiality (privacy)

The purpose of message confidentiality in authentication phase here is to hide the identity of smartphone and possible delivered secrets from eavesdroppers.

Recall from Section 2.5, that IMSI is sent in clear during the start phase of 802.1X authentication and that is a privacy issue, because TMSI, which hides IMSI cannot be used before a session has been set up.[15, p.66]. This can be compared to regular GSM network identity revealing: IMSI catching is a concept of listening radio network for phones that are powered on and register themselves to operator via GSM network.

The fault lies there, that GSM specification does not require the mobile network to authenticate itself to the phone and so GSM allows man in the middle attack device called IMSI-catcher to fake as being a base station. When the smartphone tries to attach to a fake base station, the smartphone reveals its IMSI number. Further, because the base station is responsible for chosen encryption, the base station can order the phone to not encrypt traffic at all or to use only weak encryption thus revealing all data, calls, and texts. Mitigation for IMSI-catching would be to disable GSM (2G) usage altogether from phone if that is possible.[37].

After first full authentication, client and Authenticator know TMSI and can use it in further communication: Authenticator is responsible to convert TMSI to IMSI if it later needs to ask for full authentication from the MNO.

imsi-tmsi.png

Figure 6.1 *Using TMSI*

If SIM is used as the only EAP without EAP-PEAP, then there is no mitigation for revealing the IMSI on the first message and it leads to privacy issue.

Most EAP methods do not provide identity protection themselves. Protected versions use separate inner and outer identities and that can be achieved with PEAP (Protected EAP) or TTLS, which chains different EAP-methods together and protects the inner EAP with an outer EAP. For example EAP-MSCHAPv2 (Microsoft's Challenge Handshake Authentication Protocol, version 2) can be used inside PEAP. The outer identity tells just the realm, where AuthN can be checked and inner identity reveals the real identity. The inner identity is encapsulated inside the outer identity which functions as an envelope. [TBD: speak more with protocol terms?]

Used method to authenticate depends on the inability to fake IMSI. EAP-SIM would provide identity protection, if it were used together with PEAP which protects the outer identification and then EAP-SIM were used in inner authentication. Currently it is not known for the author that implementations exists for EAP-SIM except Tseng's proposition[39] for new EAP type EAP-USIM, which extends EAP-TLS type.

If it were possible to use anonymous identity on outer EAP authentication, then EAP-SIM AuthZ must also be done at HLR AuC. AuthZ cannot else be connected to the corresponding identity and AuthN itself is not enough because it only defines the users' authenticity, not their admin roles and so AuthN should work for any smartphone that has existing contract with their MNO. It still is the responsibility of the Authenticator to check AuthZ and let only admin mobile access the management network.

Based on those facts, EAP-SIM cannot be considered confidential for identity during first message exchanges, but later the identity can be hidden using temporal identity (TMSI). Unfortunately, the TMSI is not used in this thesis for AuthN. On the other hand, EAP-SIM protocol, as do most of the other EAP-variants, provides a secure way to generate session parameters to WPA2-session and those are not leaked outside, because they are created individually on both endpoints; at smartphone and at AP.

6.5.2 Integrity

Integrity issues were handled in RADIUS Section 2.2. Message digestion codes provide integrity for RADIUS protocol. If PEAP is used, it handles integrity through its usage of TLS[29].

6.5.3 Accessibility, DoS and Scalability

Is homenet immune against (distributed) denial of service (DoS) attacks? Besides DoS, does the solution scale up from homenet to small and middle size companies? To answer this we can remember that backends (cloud and operator) are designed for thousands or even millions of concurrent users, so they hardly are limiting factors. Instead, local Authenticator might suffer from inefficiency, which comes from processing loads[20].

Traditionally, RADIUS has used connectionless UDP protocol for its light weightiness. UDP misses reliability, but retransmission in UDP is tolerable, because user

is ready to wait several seconds for authentication to complete. Today, RADIUS can also run over TCP, which has generally more aggressive retransmission rate[27, Section 2.2.1]. On the other hand, adding an alternative UDP RADIUS server can answer faster than waiting for TCP's reliable delivery.

6.5.4 RADIUS weaknesses and strengths in limited use cases

RADIUS protocol itself is old and not very secure as of current standards(2015), because messages are not encrypted and they are transported on datagrams (UDP). Alternative RADSEC protocol uses TLS, and is backwards compatible with RADIUS protocol, so it can be used as secure RADIUS proxy such as *radproxy*[41].

RADIUS uses MD5 hashing and shared secrets. Because of the weaknesses of MD5 hashing (MD5Attack[7]), the transport needs additional protection like tunneling or IPsec. TLS can be used for encryption and its signatures for integrity checking of packet payload. RADIUS protocol itself provides some integrity checks with Message Authenticators as described in Section 4.7.

In scenario III(Figure 4.3), there was a proxying RADIUS between Authenticator and MNO. When MNO notifies Authenticator that a smartphone has been authenticated, then Authenticator (AP, functioning as a RADIUS-client) hooks that message and usually just grants smartphone the access to the network. After giving access rights, other provisioning parameters can be sent with RADIUS messages, for example session time-out, current admin user list, state of OTP list, or VLAN id.

6.5.5 Replay, Re-use, Re-auth, and brute-force challenges

Earlier in RADIUS analysis, prevention of replied messages was mentioned. Reusing the same secret in different security context is also considered bad. Mixing secrets between usage domains weakens them. In GSM networks, IMSI identifies subscriber on first contact, later TMSI is used for call and SMS. In EAP-SIM those values are also used. IMSI naturally is the same, but TMSI should be different for call and EAP. Haverinen[16] explains how special RAND numbers can be used to differentiate the use of TMSI in 3GPP and LAN contexts.

Re-authentication and termination can bring unexpected results. If SSID changing introduced in mitigation Section(6.5.6) was in use, fast re-authentication should be forbidden[5, p.11]. Even, when sessions can be terminated, the client side have option to login automatically, transparent and without users control. Automatic

re-authentication after disconnection must be considered here as harmful as well as automatic login. For example, Swiss mobile operator Swisscom provided two networks for its customers: "Mobile" and "Mobile Eapsim". The latter network did not ask customers for connection and used smartphones' SIMs automatically. Unfortunately, it also charged users for using Wi-Fi connections without their knowledge.[22]

If one can read and write data through SIM card's API, one could try to get information (SRES, K_c) by brute-force. Fortunately SRES and K_c are never sent in clear, but inside a digested MAC. Additionally SIM card can be programmed to answer only limited number of challenge request, for example 65535, which in normal usage would be enough, but in brute-force challenges it would soon be exhausted and not function anymore.

6.5.6 Mitigation methods

To mitigate risks for radio capturing, two methods are presented: hiding of wireless network and proximity. They are not perfect but can limit attack vectors in time and place.

Recall that the access to management network from the smartphone is needed only then when changes are challenged. Why then not just enable management radio network during those times? Then there were less networks for users to choose from. Enabling management network could be programmed through LuCI-interface, which is a web user interface to the Unified Configuration Interface used in OpenWRT routers. Preliminary tests showed, that activating new networks in AP also disconnects existing Wi-Fi connections and may even restart AP, which certainly would not be wanted. Some other methods need to be invented to avoid denial-of service, when intruder tries to connect by that method and causes continuous AP outages.

One could also think of hiding the network by disabling the advertisement of management network SSID. That is called "network cloaking". Smartphone would then need to know the exact target SSID name. Does disabling or hiding the management network bring real security or is it just security by obscurity? Security by obscurity means here, that hiding network as a security method would filter out only some casual crackers, while at the same time it still is trivial for any serious crackers. Disabling or hiding merely gives one security layer more so it is not a real security method.

The SSID could also be renamed always, in essence to implement one-time-only network, but then the smartphone would need to get that secret somewhere, perhaps via an SMS and that would defeat the purpose of easy access. If that method nevertheless would be used, then hiding the one-time-only network name actually could add security. If the client knows beforehand the name of SSID (and maybe also checks AP's MAC), then AP does not reveal any information, before the client has tried to connect to it and that would minimize the time window for attacks.

Hiding can also have privacy enhancing effects. While Wi-Fi client's normal action is to probe for SSIDs of lately visited and learned APs, analyzing those probes anyone can reveal client's earlier locations without further effort. Lindqvist's study[21] presents usage of hidden APs in protecting privacy of clients and preventing that scenario.

Regarding boundaries of homenet, the Wi-Fi coverage gives one natural limit, which is 50 meters indoors or 100 meters outdoors, when no extenders (i.e. Wi-Fi repeaters) are in use. Proximity so brings a minimal extra layer just like network cloaking for preventing attacks, as the attacker must be physically within those limits.

This can be considered as an added factor in multifactor authentication or reputation, but it will not be enough, because attackers will have more sensitive radios available than normal users devices have. Also, if SIM-profile were used through Bluetooth, there were also range limits, but even shorter. Despite the claimed distance limits on Bluetooth, receiving can be extended with even to one mile with directed antenna[**?**].

6.5.7 Decision point for adding role information [move to design part]

Email (2014/Sep) from Karri Huhtanen revealed another problem (translation by author):

> "It is possible to add authorization message in-flight in to the ACCESS-ACCEPT. Problem is only that, if it is done in flight, you need some way to combine authentication messages to same identity. SIM auth makes it possible to use for example temporary identity and then the only thing what you can mine from the authentication message is the used operator."

So proxying RADIUS server cannot know for sure anything but the originating server (operator) if TMSI is used. The Authenticator does know the original user, but needs to get AuthZ information. It can get AuthZ either from the remote operator which would be easier for the Authenticator or there might be a proxying RADIUS, which inserts that knowledge into ACCESS-ACCEPT packet. The latter has issues with temporal identities.

When proxying RADIUS gets the temporary SIM-identity (TMSI) instead of a beforehand known IMSI identity, there will be problem on inserting the admin role information in RADIUS message. Operator or proxying RADIUS does not necessary know about roles without BaaS, so a link is needed between them to get role information inside RADIUS packet. It seems, that AuthZ data must be mapped during the first phase of EAP-SIM AuthN, when IMSI still is available, and in some way that map must be forwarded to the proxying RADIUS servers. These issues are fully avoided only in that scenario presented in Chapter 4, where there is local Authentication server in homenet. Partly avoidance can be reached, when only Full Authentication is used, i.e., authentication is always checked from MNO and no fast re-authentication is used.

6.6 Discussion

[don't jump in with so short flash messages. This looks like conclusion.] The environment is modern complex home network management. Configuration management tools are external in the cloud. Trust between homenet and cloud is searched. Smartphone lies in the intersection of both domains and possess properties to simplify binding of that trust. SIM card of smartphone, used together with Wi-Fi access to homenet verifies change controls. For verification, there are few options presented.

Location of AuthN and AuthZ components may also vary. Always in the beginning, AuthN lies outside homenet, but later it can use local point. AuthZ may be located more freely. RADIUS directs user into own virtual LAN segment (VLAN), and there management of homenet devices is allowed. That procedure activates the management port as 802.1X standard specifies. Thesis thus uses old, yet simple method for problem risen in modern environment homenet.

Disconnection from normal (Wi-Fi) access network happens, before phone can get into management network. It means, that all stateful network connections using Wi-Fi will close at that point. Smartphones do not have multiple wireless connections,

but mobile data connections may stay up. Even then, the default routing in the smartphone may change.

In the theory chapter it was questioned whether proxying RADIUS server can read and alter messages on their way or is the messaging secured by encryption, integrity hashes and digital signatures. Later it was learned, that message's integrity is protected but not encrypted.

EAP does by definition only AuthN part although successful authentication often precedes ad hoc AuthZ if nothing is demanded. EAP-SIM handles this part, but for AuthZ something else is needed and so some methods has been presented to add right role to authenticated identity.

There are many attributes in RADIUS vocabulary, which could be used to carry extra information in AuthZ phase. Exactly what of them is used remains to implementer's decision.

Regarding provisioning, it can mean adding users to home with correct attributes including authentication method and identification. There pre-existing binding between user and SIM card is carried out already. It also can mean later identifying users and giving them dynamically more attributes and access rights.

[compare with future usage, such as embedded SIMs, and past technology failures WiMax, which had no technical problems, but wifi and good cellular networks overtook it.] One motivator to this work is to find why SIM-based methods are not in wider use. The technology has been there for more than ten years and the hardware and the applications mostly support it, but it still is not yet widely used. What prevents us using the SIM?

7. CONCLUSION

Homenet's future needs, for example the change of authority and the delegation of configuration management have been described. A method to approve changes indirectly has been proposed. The approval follows from successful authentication and authorization with EAP-SIM method by the smartphone and that also sets a trust anchor to the smartphone.

For testing purposes, a real working EAP-SIM test bed with fake credentials and fake mobile operator representing EAP-SIM authentication flow was built. A dual-role model, which binds a smartphone to the homenet and grants rights to make changes, has been proposed. An indirect way to approve changes is achieved by binding the authorized access to the management network.

Complexity of existing models in interworking was one motivator for the work. Research work on the subject did reveal some of the reasons for the complexity, that are difficult to overcome with simplistic methods without at the same time losing security. There are some obvious weakness in the proposed solution. Possible usage must carefully check the safety limits even when RADIUS protocol still has strengths in security today. The thesis only scratches bootstrapping problems and issues in bootstrapping the homenet need to be studied more thoroughly. One could use tickets in Kerberized way as in GBA. Software implementation as an application is needed to the smartphone.

With the proposed technique, provisioning of users at homenets would minimize, as users already own an identifiable object, smartphone. As a positive side effect, two-factor AuthN strengthens existing security. Developing HS2.0 a few steps further would bring mobile internet off-loading on Wi-Fi networks and that would be the missing link in interworking between two worlds.

BIBLIOGRAPHY

[1] B. Aboba, D. Simon, and P. Eronen, “Extensible Authentication Protocol (EAP) Key Management Framework,” RFC 5247 (Proposed Standard), Internet Engineering Task Force, Aug. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5247.txt

[2] B. Aboba and J. Vollbrecht, “Proxy Chaining and Policy Implementation in Roaming,” RFC 2607 (Informational), Internet Engineering Task Force, June 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2607.txt

[3] J. Arkko, A. Brandt, O. Troan, and J. Weil. (2014, Oct) "ipv6 home networking architecture principles". [Online]. Available: https://datatracker.ietf.org/doc/rfc7368/

[4] J. Arkko and H. Haverinen, “Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA),” RFC 4187 (Informational), Internet Engineering Task Force, Jan. 2006, updated by RFC 5448. [Online]. Available: http://www.ietf.org/rfc/rfc4187.txt

[5] J. Arkko, V. Lehtovirta, and P. Eronen, “Improved extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA’),” RFC 5448 (Informational), Internet Engineering Task Force, May 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5448.txt

[6] M. Behringer, M. Pritikin, and S. Bjarnason. Bootstrapping trust on a homenet. [Online]. Available: http://tools.ietf.org/id/draft-behringer-homenet-trust-bootstrap-02.txt

[7] M. Chiba, G. Dommety, M. Eklund, D. Mitton, and B. Aboba, “Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS),” RFC 5176 (Informational), Internet Engineering Task Force, Jan. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5176.txt

[8] A. DeKok and A. Lior, “Remote Authentication Dial In User Service (RADIUS) Protocol Extensions,” RFC 6929 (Proposed Standard), Internet Engineering Task Force, Apr. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc6929.txt

[9] A. DeKok, “COMP128 implementation in FreeRADIUS,” accessed: 2015-05-10. [Online]. Available: https://github.com/FreeRADIUS/freeradius-server/blob/master/src/modules/rlm_eap/libeap/comp128.c

[10] W. Diffie and M. E. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, 1976.

[11] "Chapter 3 - communication security: Remote access and messaging," in *How to Cheat at Securing Your Network*, ser. How to Cheat, I. Dubrawsky, Ed. Burlington: Syngress, 2007, pp. 65 – 104,75. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B978159749231750006X

[12] "GSM - 28C3 public wiki," accessed: 2015-05-10. [Online]. Available: https://events.ccc.de/congress/2011/wiki/GSM

[13] S. Hartman, T. Clancy, and K. Hoeper, "Channel-Binding Support for Extensible Authentication Protocol (EAP) Methods," RFC 6677 (Proposed Standard), Internet Engineering Task Force, July 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6677.txt

[14] J. Hassell, *RADIUS*. O'Reilly, Oct 2002.

[15] H. Haverinen and J. Salowey, "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)," RFC 4186 (Informational), Internet Engineering Task Force, Jan. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4186.txt

[16] H. Haverinen, "Interworking between wireless LAN and GSM/UMTS cellular networks: Network access control, mobility management and security considerations," Ph.D. dissertation, Tampere University of Technology, 2004.

[17] IEEE, "IEEE 802.1: 802.1X-2010 - Revision of 802.1X-2004," 2010, accessed: 2015-05-10. [Online]. Available: http://www.ieee802.org/1/pages/802.1x-2010.html

[18] Infocomm Development Authority of Singapore, "SIM-based connection guide," accessed: 2015-05-10. [Online]. Available: http://www.ida.gov.sg/Infocomm-Landscape/Infrastructure/Wireless/Wireless-at-SG/For-Consumer/SIM-based-Connection-Guide

[19] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm," RFC 6206 (Proposed Standard), Internet Engineering Task Force, Mar. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6206.txt

[20] S.-H. Lin, J.-H. Chiu, and S.-S. Shen, "Authentication schemes based on the eap-sim mechanism in gsm-wlan heterogeneous mobile networks," in *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*, "Aug" 2009, pp. 2089–2094.

[21] J. Lindqvist, T. Aura, G. Danezis, T. Koponen, A. Myllyniemi, J. Mäki, and M. Roe, "Privacy-preserving 802.11 access-point discovery," in *Proceedings of the Second ACM Conference on Wireless Network Security*, ser. WiSec '09. New York, NY, USA: ACM, 2009, pp. 123–130. [Online]. Available: http://doi.acm.org/10.1145/1514274.1514293

[22] F. Maissen, "Kostenfalle für Swisscom-Kunden," accessed: 2015-05-04. [Online]. Available: http://www.srf.ch/konsum/themen/multimedia/kostenfalle-fuer-swisscom-kunden

[23] J. Malinen, "Linux WPA/WPA2/IEEE 802.1X Supplicant," accessed: 2015-05-07. [Online]. Available: http://w1.fi/wpa_supplicant/

[24] "EAP - Moonshot Wiki," accessed: 2015-05-10. [Online]. Available: https://wiki.moonshot.ja.net/display/Moonshot/EAP#EAP-HowMoonshotusesEAP

[25] M. Nakhjiri and M. Nakhjiri, *AAA and Network Security for Mobile Access - Radius, Diameter, EAP, PKI and IP Mobility.* Wiley, 2005.

[26] K. Narayan and D. Nelson, "Remote Authentication Dial-In User Service (RADIUS) Usage for Simple Network Management Protocol (SNMP) Transport Models," RFC 5608 (Proposed Standard), Internet Engineering Task Force, Aug. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5608.txt

[27] D. Nelson and A. DeKok, "Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes," RFC 5080 (Proposed Standard), Internet Engineering Task Force, Dec. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5080.txt

[28] OpenGroup. [Online]. Available: https://collaboration.opengroup.org/projects/jericho/

[29] A. Palekar, D. Simon, J. Salowey, H. Zhou, G. Zorn, and S. Josefsson, "Protected eap protocol (peap) version 2," Oct 2004. [Online]. Available: http://tools.ietf.org/id/draft-josefsson-pppext-eap-tls-eap-10.txt

[30] W. R. Parkhurst, *Cisco router OSPF.* McGraw-Hill, 1998.

[31] F. Pereniguez, G. Kambourakis, R. Marin-Lopez, S. Gritzalis, and A. F. Gomez, "Privacy-enhanced fast re-authentication for eap-based next generation network," *Computer Communications*, vol. 33, no. 14, pp. 1682–1694, 9/1 2010.

[32] M. Pritikin, M. Behringer, and S. Bjarnason, "Bootstrapping key infrastructures," Tech. Rep., 2014, accessed: 2015-02-04. [Online]. Available: http://tools.ietf.org/id/draft-pritikin-bootstrapping-keyinfrastructures-01

[33] H. Rachidi and A. Karmouch, "A framework for self-configuring devices using tr-069," in *Multimedia Computing and Systems (ICMCS), 2011 International Conference on*, April 2011, pp. 1–6.

[34] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865 (Draft Standard), Internet Engineering Task Force, June 2000, updated by RFCs 2868, 3575, 5080, 6929. [Online]. Available: http://www.ietf.org/rfc/rfc2865.txt

[35] B. Silverajan, J.-P. Luoma, M. Vajaranta, and R. Itäpuro, "Collaborative cloud-based management of home networks," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on.* IEEE, 2015, pp. 786–789.

[36] "Smartjac TEST (U)SIM card webshop," accessed: 2015-05-10. [Online]. Available: http://www.smartjac.biz/webstore/samples-to-order/smartjac-test-sim-configurable-options

[37] D. A. Sokolov, "Digitale Selbstverteidigung mit dem IMSI-Catcher-Catcher," *C't*, Aug 2014, visited April 2015. [Online]. Available: http://heise.de/-2303215

[38] C. Sriharsha and S. Sandhya, "Role of DIAMETER stack protocol in IMS network architecture," *IJITR*, vol. 3, no. 3, 2015. [Online]. Available: http://ijitr.com/index.php/ojs/article/view/642

[39] Y.-M. Tseng, "USIM-based EAP-TLS authentication protocol for wireless local area networks," *Computer Standards Interfaces*, vol. 31, no. 1, pp. 128 – 136, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920548907001213

[40] S. Turner and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms," RFC 6151 (Informational), Internet Engineering Task Force, Mar. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6151.txt

[41] S. Venaas, "radsecproxy," accessed: 2015-04-22. [Online]. Available: https://software.uninett.no/radsecproxy

[42] R. Ward and B. Beyer, "Beyondcorp: A new approach to enterprise security," *;login:*, vol. 39, No. 6, pp. 6–11, 2014. [Online]. Available: http://static.googleusercontent.com/media/research.google.com/fi//pubs/archive/43231.pdf

[43] J. Wey, J. Luken, and J. Heiles, "Standardization activities for IPTV set-top box remote management," *Internet Computing, IEEE*, vol. 13, no. 3, pp. 32–39, May 2009.

[44] Wireshark community, "WLAN (IEEE802.11) capture setup." [Online]. Available: https://wiki.wireshark.org/CaptureSetup/WLAN

[45] T. Xie, F. Liu, and D. Feng, "Fast collision attack on MD5." *IACR Cryptology ePrint Archive*, vol. 2013, p. 170, 2013.

APPENDIX A. SCRIPTS, CONFS, AND LOGS

[Appendices are purely optional. All appendices must be referred to in the body text, remember this!]

A.1 shell, logging options

```
#!/bin/sh -x
# Shell to start programs needed to demonstrate EAP-SIM authenticat
# on environment, where PHONE and HLR AUC are simulated.
# Used programs, all from wpa (v2.3) reference package
# - wpa-supplicant
# - RADIUS-server
# - HLR-AuC
# External WPA2-RADIUS AP hw used
#
# options:
# -t more timestamps to xxx...
# -K include keydata to debug
# -dddd more debug
#
# usage:
#       ./apd [OPTION]...

# root directory for programs and logs
BASE=/home/itapuro/gitdocs/di/testit

# Client (supplicant) parameters
WPASUPPLICANT=$BASE/wpa_supplicant
# hostapd as RADIUS role NOT AP-role
HOSTAPD=$BASE/hostapd
# Mobile operator (Home location register authentication centre)
HLR=$BASE/hlr_auc_gw
# if only cred part is in, does not work
WPASUPPLICANTCONF=$BASE/wpa-simtest-owrt2.conf

# HLR_AUC_GW parameters
# sim triplets, when EAP-SIM used
```

```
SIM=$BASE/hostapd.sim_db
#  Milenage parameters, when AKA used
MILENAGE=$BASE/hlr_auc_gw.milenage_db

# HOSTAPD parameters
# settings for hostapd include wired, eap_server, eap-handler
HOSTAPDCONF=$BASE/hostapd-jmdemo.conf

# timestamped logs and  confs into safe
TIMESTAMP=`date +s%y%m%d-%H%M%S`
TARGET=$BASE/demot/ap-$TIMESTAMP
mkdir $TARGET
cp $0 $HOSTAPDCONF $SIM $MILENAGE $TARGET
# reset programs, if still running.
pkill hlr_auc_gw; pkill wpa_supplicant; pkill hostapd
# Killing does not clean up some locks and sockets
if [ -S /tmp/hlr_auc_gw.sock ] ; then
  rm -f /tmp/hlr_auc_gw.sock
fi
if [ -S ./eth0 ] ; then
  rm -f ./eth0
fi

## 1. HLR_AUC
# startup using SIM-triplet
# $HLR -g $SIM  >  $TARGET/hlr-debug &
# startup using MILENAGE. Works also with SIM
$HLR -m $MILENAGE > $TARGET/hlr-debug  &

### 2. HOSTAP (in RADIUS-EAP-handler mode)
# initialization
ifconfig wlan0 up
# captures for RADIUS (wired, AP-RADIUS) and
# EAP (wireless, client-AP)
tshark -i eth0  -w $TARGET/eth0-pcap &
tshark -i wlan0 -w $TARGET/wlan0-pcap &
echo start hostapd
# & pakollinen, jos >
$HOSTAPD -Kdt $HOSTAPDCONF > $TARGET/hostapdwired-debug  &
```

```
echo "if␣$?␣==␣0␣then␣RADIUS␣server␣started␣ok"
sleep 1

### 3. WPA_SUPPLICANT
echo starting supplicant..
$WPASUPPLICANT -dK -iwlan0 -c $WPASUPPLICANTCONF \
        -D nl80211 > $TARGET/wpasupp-extapradius-debug &
### Live analysing
echo starting analyze..
cd  $BASE/demot
cd `ls -d ap-*|tail -1`/
sleep 1
# follow 3 log files, with color coding set in multitail.conf
multitail -F ../multitail.conf -N 10000  -CS eap-sim -ts host*debug
wpa*debug

# alternatively, start this in own window
# xterm -e $BASE/demot/anamulti &

# tests won't take 15 mins..
sleep 900
# if they did, somebody had fallen in sleep.  Commit logs.
pkill tshark
git add $TARGET
git commit -m "apd-tty␣tests␣$TIMESTAMP␣"
```

A.2 wpa-supplicant settings (wpa-simtest-owrt2.conf)

[already in text] If KEYS have been excluded from log files, there as placeholder stays string "[REMOVED]".

```
# EAP-SIM with a GSM SIM or USIM
beacon_int=10
network={
        ssid="simtest"
        key_mgmt=WPA-EAP
        eap=SIM
#       pin="1234"
#       pcsc=""
  identity="1232010000000000"
```

```
  password="90dca4eda45b53cf0f12d7c9c3bc6a89:cb9cccc4b9258e6dca4760
}

cred={
        imsi="1232010000000000"
        milenage="90dca4eda45b53cf0f12d7c9c3bc6a89:cb9cccc4b9258e6d
}
```

A.3 RADIUS server conf

```
# no wireless functionality, only RADIUS/EAP
driver=none
# RADIUS secrets for external AP
radius_server_clients=hostapd.radius_clients
# eap-handler enabled
eap_server=1
# mapping of eap credentials to SIM,AKA and AKA' protocols
eap_user_file=./hostapd.eap_user
# Inter-process communication with hlr_auc_gw process
eap_sim_db=unix:/tmp/hlr_auc_gw.sock
```

A.4 hlr auc

A.5 No sim

Here capture + analysis from nosim

B. [MISC TO BE ADDED ON RIGHT PLACES]

B.1 facts TBD.

- "most EAP authentication protocols lack two features: identity protection and withstanding man- in-the-middle attacks. "

source :

Yuh-Min Tseng Department of Mathematics, National Changhua University of Education, Jin-De Campus, Chag-Hua City 500, Taiwan, ROC.

"USIM-based EAP-TLS authentication protocol for wireless local area networks" and

Wireless (In)Security www-page, where EAP table shows that PEAP has MiTM. `http://networking.ringofsaturn.com/Security/WirelessInSecurity.php`

- re-auth for long-lived sessions or if there is cost for disrupting them
- APs provide different authentication suites for different

SSIDs

```
essid="nurkka"
          IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
essid="simtest"
          IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : 802.1x
```

B.2 using EAP for other than network access, i.e., for application auth.

- `http://www.rfc-editor.org/rfc/rfc7057.txt`
  - EAP
- application as an EAP peer
- RFC6677: Channel-Binding Support for Extensible Authentication Protocol (EAP)
- channel binding must be used

B.3 eap-psk rfc4764.txt

- other limitations than identity protection are password support and Perfect Forward Secrecy (PFS).
- eap-psk
- only 3 standards track EAP methods per IETF terminology,

but all of them are deprecated (md5,OTP,GTC ja?)

- some EAP- o Essentially require additional infrastructure, e.g., EAP-SIM [1], EAP-AKA [2], or OTP/token card methods like [3].

B.4 eap-sim acts similar than any other EAP challenge method (or not?)

- compare eap-sim with other method and point out differences.
- privacy already shown
- user defined passwd?
- how many messages needed? In eap-sim 6+1 (success/failure)

[1] DEFINITION NOT FOUND.
[2] DEFINITION NOT FOUND.
[3] DEFINITION NOT FOUND.

- to do comparison, I have to study how EAP in general works
- Authenticator for example can use either local method or pass-through the authentication to external backend, still keeping EAP-message in tact(sp?) as of rfc4137
- WPA2 package's hostapd from JM does not perhaps provide EAP-PEAPv0 SIM but

wpa-supplicant supports.

- VLAN itself has an attack vector and some methods exists, but also mitigation for them.
- rad Authenticator, TLS, RADSEC etc, needs both client&server in x509 certificate