RIKU ITÄPURO

SMARTPHONE AS A TRUST ANCHOR IN HOME NETWORKS

draft-11.5.2015 Master of Science thesis

RIKU ITÄPURO

SMARTPHONE AS A TRUST ANCHOR IN HOME NETWORKS

draft-11.5.2015 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-11.5.2015 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.]

Existing work done at TUT for delegated homenet configuration currently has preliminary authentication and access model using pre-defined credentials and SSH-connection from controller device to configuration targets. It misses the bootstrap of infrastructure i.e. the first trust. Smartphone with its SIM card and existing key infrastructure to mobile network operator eliminates the need for additional credential distribution. It is discussed and shown how mobile authentication is done using extendable authentication profile (EAP) with SIM-card.

Theory, how SIM-authentication works is presented and simulated environment to demonstrate that is built, tested and analyzed. As a result it is shown, that SIM authentications 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 area as homenet and delegated management. For transporting authentication claims, WPA enterprise has been chosen, which includes RADIUS environment. To further avoid complexity and granularity, we only use simple model of management network. Getting in to management network is carried out at homenet 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. 3

2.1 802.1X 4

2.2 RADIUS 4

2.3 WPA 5

2.4 EAP 5

2.5 SIM-based authentication 6

2.6 Trust 8

2.7 Need for Security bootstrapping -> in Chapter 9

3. Managing Home Networks [or Home network architecture] 10

3.1 Home network architecture and IETF 10

3.2 Centralization trends in management 12

4. Design of home network trust anchor 13

4.1 Alternative methods for introducing trust anchor into the homenet . . 13

4.2 Flow of design (already above) 15

4.3 Chosen design and why (Rationale) 18

4.4 Access methods to Wi-Fi with only one SSID 19

4.4.1 HS2.0 [If deleted, remember also from conclusion! TBD] 20

4.5 Scenarios for authorization (AuthZ) 21

4.6 Ways to modify RADIUS messages 25

4.7 Privacy of smartphone user's identity (IMSI) [-> to secur. on cha6] . 26

4.8 Similarities with Lock-and-Key method 28

4.9 Chosen solution 28

5. Implemented Solution 30

5.1 EAP-SIM authentication test bed 30

5.2 Detailed description of test runs 31

5.3 Disconnecting the local controller and offline changes 31
5.4 Network traces (EAP, SIM, AUTH traffic analysis) 33
6. Analysis, Results and Discussion 37
6.1 Deployment difficulty 37
6.2 Estimating time to authenticate EAP-SIM 37
6.3 Costs for end-user 37
6.4 Platform specific issues 38
6.5 Security considerations 39
6.6 Accessibility, DoS and Scalability 40
6.7 RADIUS weaknesses and strengths in limited use cases 40
6.8 Replay, Re-use, Re-auth, and brute-force challenges 41
6.9 Mitigation methods 41
6.10 Discussion 42
7. Conclusion 44
Bibliography 48
APPENDIX A. Scripts, confs, and logs 49
A.1 shell, logging options 49
A.2 wpa-supplicant creds 51
A.3 RADIUS server conf 52
A.4 hlr auc 52
A.5 No sim 52

LIST OF FIGURES

2.1 EAP-logical layering 6

2.2 EAP-SIM full authentication sequence diagram, based on RFC4186 . 8

4.1 Scenario I with 3 separate domains: BaaS, MNO and homenet 22

4.2 Scenario II with AuthZ in homenet 24

4.3 Scenario III with outsourced AA 24

4.4 Scenario IV, AuthZ from BaaS, AuthN from homenet 25

5.1 EAP-SIM AuthN messaging in simulation testbed 30

LIST OF TABLES

2.1 Comparison of WPA-PSK and WPA-ENTERPRISE modes 5

2.2 Setup tasks in WPA2-Enterprise with EAP-PEAP-MSCHAPv2 and EAP-SIM 7

4.1 Location of AA, AuthN and AuthZ in scenarios I-V 21

LIST OF PROGRAMS

./testit/apd–tty.clean 49

testit/wpa–simtest–owrt2.conf.clean 51

testit/hostapd–jmdemo.conf.clean 52

LIST OF ABBREVIATIONS AND SYMBOLS

TUT	Tampere University of Technology
URL	Uniform Resource Locator
3GPP	3^{rd} Generation Partnership Project
AAA	Authentication, Authorization, Accounting
AKA	Authentication and Key Agreement, used in 3GPP mobile networks
AUC	AUthentication Center
CPE	Customer Premise Equipment, device physically located at customers home.
EAP	Extensible Authentication Protocol, extends 802.1X
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.

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)]

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 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

Earlier it was sufficient to make just minimal settings at home to modem (cable, phone or radio) and connect it to the home computer. That enabled fully working home network with Internet connectivity. Now homenet 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 beside computers and printer.

Connecting these to the net have become more complex, even at home. Persons, who are allowed to make configuration changes are often limited only by simple password authentication and physical precence near configured network device. Sensor and controller devices from Internet of Things domain bring their own increment to the device count and their owner may not necessarily be the same than the home owner so there will be a need to delegate management of homenet to multiple owners.

There will be a market for an external consultant service, which could remotely operate the homenet for example through service in the cloud.

The service provider therefore does not need to have direct data access to home but only to the cloud based service for to be able to manage homenet devices. Consultant service is not the only possible delegation for homenet. Homenet can be divided to multiple segments that each have their own administrative parties. For example electricity company may have sensor and controller network physically inside the homenet, but logically separated from other parts of homenet. It is then important to keep track of who is allowed to access which part of the homenet. Eliminating the need for physical precence at home reduces external actors' costs, but adds many questions regarding security, not only to overcome firewalls, NATs, and disconnections.

Those security issues must be solved before delegation in the cloud can happen. One of them is the authentication and authorization from the cloud to homenet. To secure the connection from cloud service (controller) to the homenet, 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 homenet.

Any encryption between devices needs trusted key exchange beforehand and finding and establishing trust is needed for that. Human aspect and usability also are important. Proposed model should take people less effort than current methods of distributing user credentials, finding the place where they can be inserted, and ensuring that they are written correctly. [write more clear and mention key distribution problem.] The place where trust is not anymore derived or built upon other fact, but assumed to be present is called a trust anchor.

Main focus is on authentication and authorization part of homenet management with smart phone as trust anchor. Trust model, which benefits from smartphone's unique, existing secret keys inside smart card Subscriber Identify Module (SIM) is proposed. SIM here means keys and application in SIM or USIM inside UICC card. Besides that, problems such as limited connectivity are studied.

Why SIM-based methods are not in wider use is one motivator to this work. The technology has been there for more than ten years and hardware and applications already support it, but it still is not yet widely used. Could there exist a light method to use SIM? Combining existing techniques, thesis presents one possible way to bind the trust to the 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.

Thesis is divided as follows: Chapter 2 explains Authentication-Authorization model. Chapter 3 describes security in current homenet architecture and current practices for configuring it. Chapter 4 discusses methods to bring trust anchor in the homenet and explains the chosen method. One specially crafted problem is how any scenario presented 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.

[delete items after paragraphs ready]

- 1) Different technologies for access control, authentication, authorization

1.5) wireless (authenticator, authentication server, supplicant)

- 2) RADIUS, diameter, (tacacs+)
- 3) SIM-based authentication

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

1. GBA and Security bootstrapping

Authentication, authorization, and accounting services (AAA) are components for access management. Of these only first two A's are used here and later described as AA services. Authentication (AuthN) answers how to identify users and proof that they really are who they claim to be. Authorization (AuthZ) answers what operations 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 object wants to access. There AuthN is checked against credentials file and authorization from service specific policy file. To be more exact, identification preceding authentication is part, where entity claims and presents its identity to access controlling system.

AA services need to trust some entity endpoint. From that point, trust can be chained all the way to the access decision point. The trust end point is called a trust anchor.

2.1 802.1X

802.1X [16] is an IEEE standard protocol for port based access control. Network access through specific port is restricted (controlled) from client (called Supplicant) before client has successfully performed AA. 802.1X device, where controlled ports are located, is called an authenticator. Third part in 802.1X is an authentication server. Authenticator may consult external RADIUS server for authentication requests.

In homenets authenticator usually is inside the access point. On large enterprise networks, authenticator can be centralized and access points function only as radio stations. 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.

2.2 RADIUS

RADIUS is the most popular provider for AAA-services [30, p.75]. It was used first for with remote terminal and dial-up modem users, hence the name Remote Authentication Dial-In User Service. Later is was used as centralized AAA for networking devices such as switches and routers. Currently its main environment at home and SMEs (Small and Medium-sized Enterprises) is wireless connections (Wi-Fi). Besides RADIUS, there exists similar protocol called Diameter which is newer than RADIUS and in use in 3GPP (4G?) networks.

Here RADIUS-server takes role of authentication server. RADIUS-protocol is stateless, request-response type client-server protocol. RADIUS messages used for ACCESS are (ACCESS-REQUEST, ACCESS-RESPONSE, ACCESS-ACCEPT, or ACCESS-REJECT). ACCESS messaging-flow includes AuthN and AuthZ. When both AuthN and AuthZ succeeds, ACCESS-ACCEPT message is sent back to Authenticator and access is granted to protected port. Besides authentication, other service parameters such as provisioning can be included in ACCESS-ACCEPT message. In essence, AuthZ part itself can be thought as one type of service provisioning. [22].

AAA-protocols don't dictate policies, ie. who is granted access or what operations user is allowed to do. They only transport these information between client and authenticator server. EAP[first mentioned here!] is used to transfer only authentication messages, instead of session keys.

If there are multiple RADIUS servers, the messages are chained and proxied always to next RADIUS server ie. proxying RADIUS server. In the following Chapters it

is discussed how proxying servers take part in AA decisions. Of main interest 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 [5].

2.3 WPA

Wireless protected access (WPA) protects traffic in wireless, shared media, where everyone can simple listen the traffic on radio waves. It allows both authenticated access and message encryption. WPA-supplicant is client software for 802.1X and communicates with the authenticator.

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

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

Property	WPA-PSK	WPA-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 modular Extensible Authentication Protocol (EAP) framework [4]. EAP has types for example for hashed passwords, TLS certificates, or SIM/AKA using smartphone's SIM card.

It must be noted that EAP tells only messaging form, so it needs to be encapsulated inside another protocol. In Wi-Fi, between smartphone and access point, EAP is encapsulated into 802.1X protocol (EAPOL) or into TLS protected PEAP (Protected EAP) [24] before sending into wire. In wired net, RADIUS encapsulates EAP messages. Encapsulation is described in Figure 2.1 and there it can be seen, that EAP messaging happens logically between EAP peer and authentication server but on lower, transport layer there is EAP authenticator in between them, which transfers EAPOL messaging into RADIUS message. In the end (not shown in the Figure 2.1) authenticator is responsible for opening access for EAP peer. This work uses EAP type of EAP-SIM.

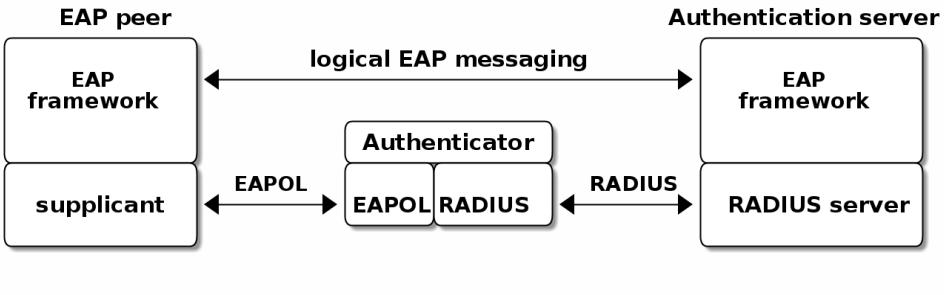

Figure 2.1 *EAP-logical layering*

2.5 SIM-based authentication

MNO and SIM trusts mutual each other. There is still need for separate access credentials for Wi-Fi and that was the reason of developing EAP-SIM and later derivatives EAP-AKA and EAP-AKA'. The goal was to combine in a secure way existing GSM keys for Wi-Fi access. Existing general purpose EAP-methods in 2004 were not compatible with GSM protocols for this purpose. [15, p.93]

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. It is defined in RFC4186 [14]. It is challenge-response method and similar to AuthN used in GSM, but EAP-SIM adds mutual AuthN ie. also the network is authenticated. In EAP-SIM, client sends a nonce, which is by definition a value used only once, and that must be received back in correct form in network's signature response.

Authentication server generates challenge with aid of triplet from MNO. That procedure is later described in more detail.

Upwards from 3GPP network, types EAP-AKA and AKA' can be used. EAP-AKA is defined in RFC4187 [7] and uses 3GPP's AKA (authentication and Key Agreement) protocol. It differs from SIM by using additionally parameters from MNO to protect replay attacks. Otherwise the protocol messaging is same as in GSM-SIM, only algorithms differ.

Last, there exists EAP-AKA' (AKA-PRIME) [8]. Enhancement to AKA is to include Service Set name (SSID) in the key derivation function. Additionally, digests use SHA-256 function instead of SHA-1.

Using EAP-SIM means using secret key inside SIM card with A3/A8 algorithms to generate valid responses for challenges coming from MNO and to derive session

keys. Algorithms used (A3/A8) and their possible implementations (COMP128, COMP128v2, COMPv3) are not of interest in this work besides the point that they are mobile operator specific or known reference algorithms. Algorithm used in demo was internal GSM-Milenage for EAP-SIM, which is a reference implementation and as such suitable for operators who do not want to invent their own security algorithms.

In many parts, SIM variants in EAP are simpler, than other EAP variants to mobile client. Table 2.2 compares setup of Wi-Fi in clients of one some existing organization compared to EAP-SIM. It is noteworthy, that plain EAP-SIM will not support identity hiding and that will be later be discussed further. If we add PEAP [24] also to EAP-SIM, comparison will be more fair. As can be seen from table, leaving certificates out from environment makes client setup easier with price of revealing smartphone user's identity.

Table 2.2 *Setup tasks 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
choose CA	x		x
tell CA to clients	x		x
if CA not known, distribute it *secure*	x		x
enable PEAP	x	N/A	x
set used EAP-method	x	x	x
set validating of RADIUS server	x		x
set encapsulation (WPA/802.1X)	x		
set outer identity	x		x
set inner creds	x		
hide identity		N/A	

Sequence diagram of full EAP-SIM authentication supplicant (here smartphone) and authenticator (in AP) is shown in Figure 2.2.

[EXPLAIN what are used] Important parameters for this work are IMSI, NONCE, and triplet values corresponding IMSI (RAND, SRES, Kc).

[Description of protocol important or not?]

Unique identifier for SIM is IMSI (International Mobile Subscriber Identity, 15 digits long, more familiar user's phone number. From the diagram we can see, that IMSI is revealed in message 2 in plain-text. Later, after session has been set, IMSI may be left out and temporal IMSI (TMSI) can be used, to hide client's identity.

All EAP-SIM derivatives provide mutual authentication. Without NONCE in message 4, that would not be possible. NONCE is by definition, once used random string or number. Client challenges the network by sending NONCE during start of the negotiation phase. It later checks in message 7 whether RAND values from operator were digested with correct NONCE.

Yet some documents claim, that EAP-SIM does not provide mutual AuthN, so what can be the case? Perhaps they mean, that mutual AuthN is not provided between mobile and RADIUS servers. Another explanation is, that in AKA and AKA' network is authenticated in very early phase with help of operator specific symmetric keys, which are also inside SIM.

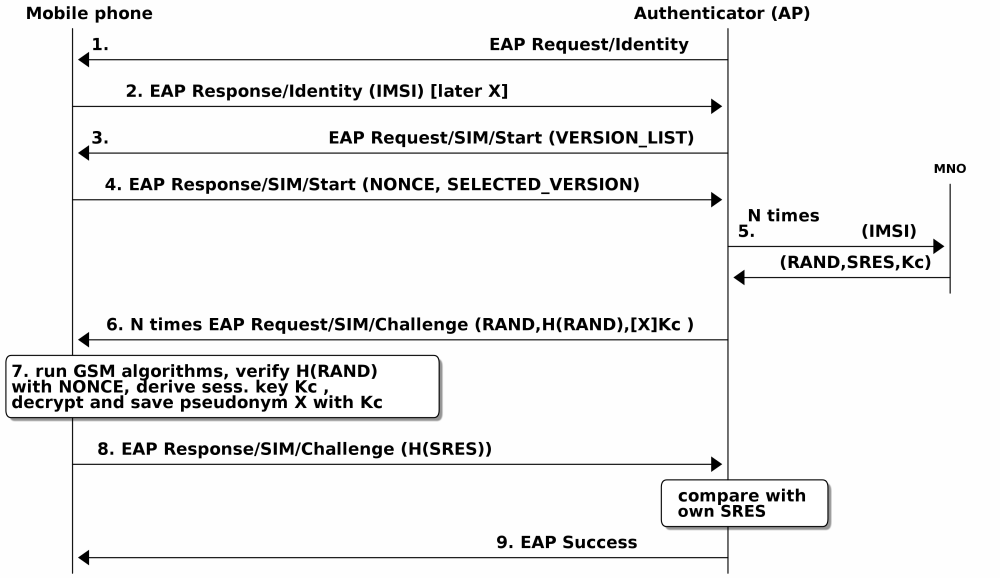

***Figure** 2.2 EAP-SIM full authentication sequence diagram, based on RFC4186*

2.6 Trust

Trust is the base. Secure communication has many layers. On its base lies trust. Without trust there is little help with any added encryption or secrecy. Setting trust is usually not an easy task, but only after completing that phase it is meaningful to complete the other security layers. For example, secret keys enable encrypted communication, but the keys need to be delivered through an trusted channel, and so it can be seen that trust really is the first layer to be fixed.

Even without trust, some form of secure key-exchange is achievable with Diffie-Hellman key-exchange [12]. Unfortunately, it is vulnerable to Man-in-the-middle(MitM)

attacks, where protocol does not notice, if messaging goes through third party, which inpersonates itself to both ends as being the corresponding messaging partner and can read encrypted messages. With trust set between two devices, ie. if they can securely authenticate each other, secret communication is possible. Secure network configuration and credential exchange is then possible.

As mentioned earlier, the smart phone and MNO trust each other hence mutual authentication between them is possible. Now, how this could be used to include other components under same trust circle in the homenet? As AuthN-AuthZ at home proceeds through authenticator, maybe authenticator can deliver this information further and use it as a derivation function to extend trust.

EAP-SIM derivatives provide strong AuthN which means here two-factor AuthN. Software certificates, while stronger than regular passwords, do not possess properties *non-copiable* or *unique*, so they can only be considered as strong passwords and they don't 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.

2.7 Need for Security bootstrapping -> in Chapter

[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?]

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 computer network located at person's home. It consists of devices and their interconnect, either wired or wireless. This thesis denotes home network as homenet, although the name 'homenet' is reserved to Internet Engineering Task Force Working Group (IETF WG) homenet. IETF is responsible for the most Internet technology standards. Current drive in homenet management is towards IPv6 environment as it allows future addressing and routing needs. As old technology cannot be forgotten, homenets will be heterogenous having both old and new technology and their interoperability is important in planning future homenets. Segmenting home in multiple subnets will belong to homenets and will include areas for home members, guests, and management.

Securing homenet and its router's configuration is done by limiting traffic with static or dynamic access control lists (ACL) in routers. ACLs in turn are secured from change by AAA. Authorized agents can make changes, either direct in the device or through some management protocol such as SNMP or NETCONF[source]. SNMP has been in use for over 30 years and well supported in routers. Only 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.

Management of devices on border of homenet and operator have been done already earlier. For example TR-069 standard exists [35] for CPEs such as ADSL broadband routers or set-top boxen. TR-069 has been used to implement self-configuration architecture in homenets [27]. On these days research is done with Light-weight Machine to Machine (LWM2M) processes.

RFC7368 about IPv6 Home Networking Architecture Principles from Arkko [6] defines the borders of the homenet and states that internal borders in homenet should possible be automatically discovered but continues by saying that limiting borders to specific interface type makes it difficult to connect different realms locally. The same document continues stating that while homenet should self-configure and self-organize itself as far as possible, self-configuring unintended devices should be avoided and let homenet user decide whether device becomes trust. So, these statements reveal us that homenet environment still needs external configuration even with proposed automation aids.

Homenet WG proposes use of Public Key Infrastructure (PKI) at the home. 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 Behringer's draft [9] proposes, that first one device is chosen for the trust anchor and trust is built upon that anchor. This anchor device then becomes homenet's Certificate Authority service. In the end, rest of the homenet will be imported into homenet through CA, which returns their certificate requests signed.

Key creation, key exchange and their usage is explained in similar draft from Pritikin[26]. There is also discussion about using manufacturer provided device certificates as trust anchor. 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. 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. [source?]

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

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 [?] for code propagation are used in homenet. Configuration also includes configuring power level setting of devices to save electricity based on usage profile. For example at nights or when there are nobody home, some devices don't need to be working at their maximum capacity. Instead, we study interface of AAs. Main points here are existing infrastructure (phones, Internet access, Wi-Fi access points), strong authentication (two-factor), and existing 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 devices console or web-access. As number of devices has increased, it would have been reasonable to rationalize the process by central management device, not least to prevent human errors for repetitive tasks. Yet, at home networks 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 the more centralized model, smartphone is set here as a central and managing local controller.

Users already have one phone, which can be considered as 'smart' and most smartphones have Wi-Fi capabilities and suitable for Local Controller between cloud and homenet. When we choose smartphone to be the management point, the other benefits are numerous: management software can be delivered and updated from cloud to diverse smartphone types, and existing user base is enormous. Operator located user databases in Home Location Registry Authentication Center (HLR-AuC) still have orders of magnitude more users available than any organization.

4. DESIGN OF HOME NETWORK TRUST ANCHOR

[Chapters contents here]

Key distribution problem is solved at SIM-card distribution phase. SIM card authentication is strong: there is physical SIM and secret PIN for it. Smartphone then belongs to same category as (intelligent) USB-dongle, RSA-ID or Secure-ID hardware devices. They are part of "what you own". Trust exists to mobile operator, and that is later shown as an important factor.

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.

The smartphone connects with Wi-Fi link to access point (AP) in homenet. AP functions there as an authenticator. Trusted connection is needed between existing network and Local Controller ie. homenet and local controller need to trust each other. Smartphone will approve changes for homenet and is part of bootstrapping new infrastructure.

4.1 Alternative methods for introducing trust anchor into the homenet

Before fully explaining our chosen method, we introduce some alternative approaches for trust anchor. Trust anchor is part of bootstrapping. Trust information, may it then be a secret or some evidence, can be delivered to trust device via physical transport. Traditional way to do that is with password inside sealed envelope or one-time password list what online banks today use. Secret can also be sent as an SMS.

Trust can be requested with help of trust anchor's unique properties. Some new devices have vendor certificates inside them which brings public key infrastructure

as possible alternative. Device presents itself with a certificate, which has been issued by a trusted vendor. Keys are then in device's trusted hardware store. Vendor-trust is needed for checking issued certificates. Root CAs, trust anchors also, can be read from device's read-only store. CPE could use vendor certificate for AuthN of earlier unknown device. If keys are stored in SIM as here, external operator support is needed.

Other techniques than EAP-SIM to use SIM's unique properties 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 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 (ie. IMSI) before it answers the call. That information is called Caller ID and it has been is 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) defines standard for mobile signature services (MSS) in ETSI TS 102 204. MNO's in Finland have implemented this as a Mobiilivarmenne service. For example, Sonera's brand for it is "Sonera ID" while Elisa calls it "Elisa Mobiilivarmenne".

When AuthN and AuthZ comes from outside, one possibility is to use 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 clients.

Project Moonshot, if worked and used together wit 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 inner protocol of EAP-TTLS. [2]

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 mobile, when access confirmation is needed? Mobile SIM provides two-way AuthN part as discussed earlier. 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, 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 key distribution problem.

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 WPA 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 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 (already above)

Wanted:

- separate MGMT net exists
- SIM authentication to MGMT net is proven
- changes are authorized if they come from MGMT net
- log-out from MGMT net

(- spare connection, if internet link down) (- fast-reauth, without MNO

Implications are, that when someone has access to MGMT channel, everything is permitted. No security limiting as default

[Basically 2. and 3. is like traditional corporate network with firewall.]

a. AuthN is proven

b. AuthZ decision has challenges

c. Change approving has three cases:

1. Changes are allowed, when port is open
2. Confirmation message from MGMT-net authorizes changes. Message must belong to configuration and can be example a digested signature.
3. FULL: changes may come only from MGMT net.

Use-case for adding 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 new actors, such as new operators, who would want to have access to separate segments of homenet. 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.]

Changes could be marked some way, so that they need approving. When CPE of homenet is about to input configuration changes which would change balance of authors or roles, it will first need to ask for permission. It does it by asking from trusted point, here mobile SIM.

[How is this PULL asking triggered? In reality it is not asked, but changes are accepted from admin roles. How admin role is checked?]

CPE wants to verify if the changes authorized. They are, if currently smartphone user is logged in management network (i.e. management is allowed). Additionally, there could be a specific change-approval message, which must be sent through management network, maybe including digest of change message as a verification and.

Because smartphone is not actively listening the CPE, how it could input that 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.
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. Smartphone would authenticate (if not already) in to management network 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 authentication token with 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 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 correct management network. That implies disconnection from other services, because smartphone currently has only one Wi-Fi radio available. It is not tested, whether 3G-data link could be active still at the same time.

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's etc). Analogy to real world would be public access corridors and doors for customers separate from privileged doors for service personnel.

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

Naturally one will first challenge the solution, if management network is thought to be in secured zone. but sure devices have additional protection for logging in them.

Example of complex solution would be a traditional firewall and packet inspection in interconnects. Even more complex would be that traffic always travels through Access Control Engine such as Google's BeyondCorp [34], where all traffic is suspected as being external, even when it originates from inside networks.

In production, some changes in controller are propagated to homenet via management network without need for extra authentication phase. 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 homenet needs secure binding to the mobile controller, earlier mentioned trust is the first one needed. The trust is achieved by checking whether the mobile controller 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 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 Extensible Authentication Protocol (EAP) which selects specific authentication mechanism [4, p.3], 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 [22, p.4]. RADIUS can bring extra attributes in its ACCESS-ACCEPT message. Specific VLAN attributes can be delivered via Vendor Specified Attributes (VSA), if standard RADIUS messages do not suffice. VSAs allow vendor to use extra 255 attributes as they wish, but also currently there exists RADIUS extensions for directing user into VLAN That way (3rd party) authentication server can divert and segment areas of home network. 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, already puts the user in to wanted network. As for other provisioning parameters, not all end devices support them.

In Behringers work-in-progress bootstrapping [9], AuthZ happens likewise first at cloud providers 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 authenticator in CPE.

4.4 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 access point. Having multiple SSIDs enable us to dedicate one of them to management network. To enable EAP-SIM method, it is necessary to use WPA-Enterprise mode an as such, to use RADIUS server.

It was not found, how authenticator could use the same network with both WPA-PSK (or open access) and WPA-Enterprise, so separate SSID for management network was technical 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 only checking comes from 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 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.

4.4.1 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)

4.5 Scenarios for authorization (AuthZ)

[Place of Authorization decision]

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

Here are presented 5 scenarios for possible locations of AuthN and AuthZ points. Authenticator is the entity which gives the final decision about access. 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.

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	-	-	-

[1] BaaS provides

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.])

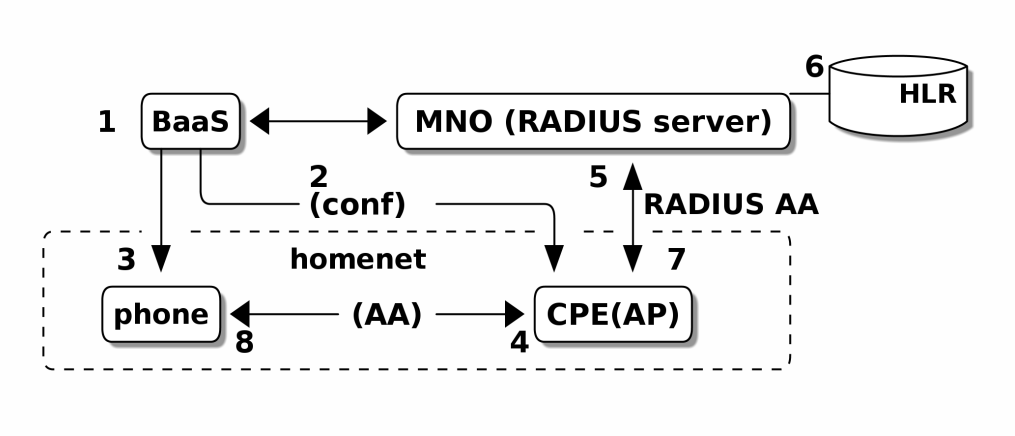

Figure 4.1 *Scenario I with 3 separate domains: BaaS, MNO and homenet*

[Maybe replace BaaS with CLOUD]

[alt. presentation of flow number I, 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). [how long does it take to ask?] MNO returns in RADIUS message either *ACCESS-ACCEPT*, if user is both known AND has admin role or *ACCESS-REJECT* (7).
7. CPE receives this ACCEPT or REJECT. If there were other RADIUSes between CPE and MNO, they would have acted as proxy RADIUS servers.

8. IF ACCEPTed, then mobile is both authenticated and authorized (8) and can send configuration change message to CPE, which recognizes it coming from authentication network.

[alt. presentation of flow number II, paragraph]

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 IMSI user with its HLR-AuC (6). MNO also asks from BaaS, whether IMSI user has admin-role (AuthZ). [how long does it take to ask?] 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 mobile is both authenticated and authorized (8) and can send configuration change message to CPE, which recognizes it coming from authentication network.

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

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

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-

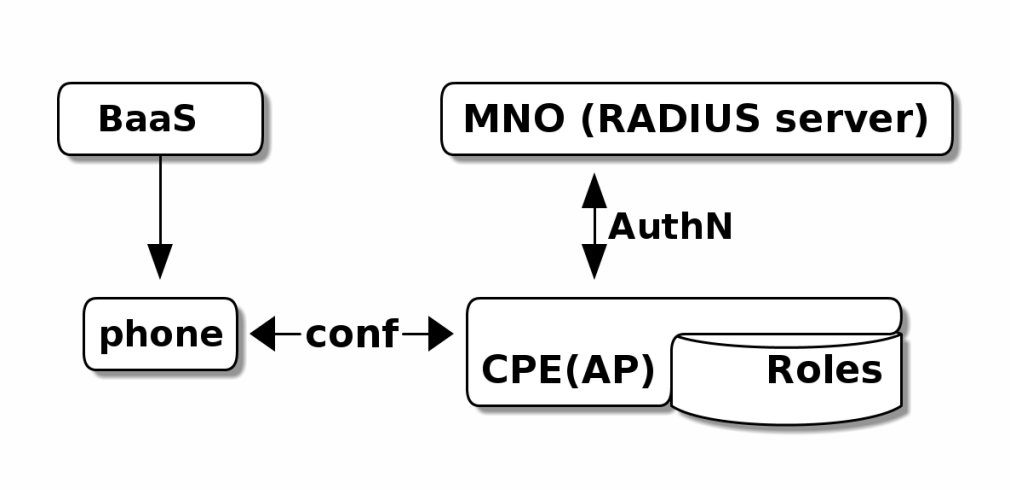

Figure 4.2 *Scenario II with AuthZ in homenet*

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 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.

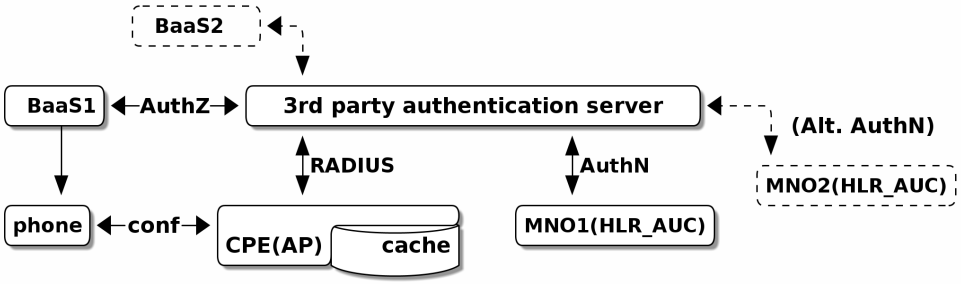

Figure 4.3 *Scenario III with outsourced AA*

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.

Scenario IV (Figure`fig:scenario-IV`) is almost like scenario II, but AuthZ is always checked from BaaS. If there are no connection to cloud, fall-back is to work as II. So also this scenario needs local store for admin IMSIs.

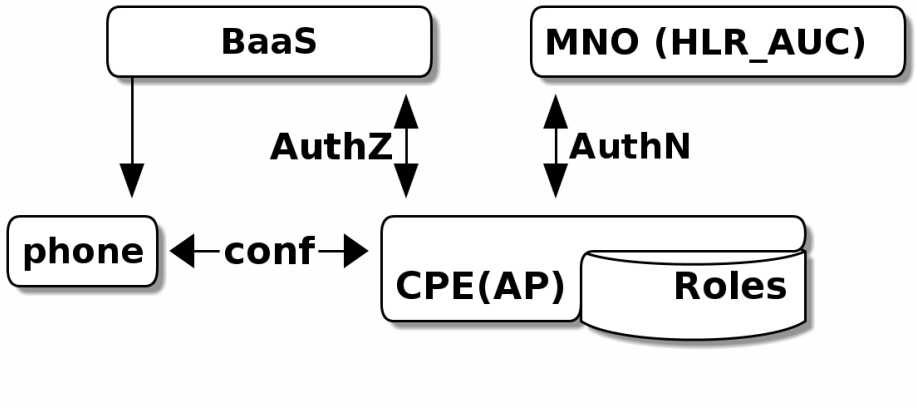

Figure 4.4 *Scenario IV, AuthZ from BaaS, AuthN from homenet*

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.

4.6 Ways to modify RADIUS messages

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. [13, 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. [13] If the user password was to be transmitted on wire, it would be run through exclusive OR function (XOR) together with MD5 digested Requst Authenticator and Secret and put in to User-Password attribute. In other words

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

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 from elsewhere.

RFC2865 [28] says, that forwarding proxy may alter the packet as it passes it. Because change invalidates the packet's signature, proxy has to re-sign it. Later, RFC6929 [**?**] reminds, that even when proxying RADIUS does not understand all AVPs inside RADIUS message, it must deliver those values. That allows us to use larger set of AVPs than is in all RADIUS servers vocabulary.

So at least Proxying RADIUS can insert something, but is that 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 [32, p.2]. MD5 can not be thought as computationally secure, because duplicate hashes are easy to compute today [37].

4.7 Privacy of smartphone user's identity (IMSI) [-> to secur. on cha6]

Recall from Section2.5, that IMSI is sent in clear during start phase of 802.1X authentication and that is a privacy issue. TMSI cannot be used before session has been set up. [14, p.66]

Most EAP methods do not provide identity protection themselves. Protection uses separate inner and outer identities and it 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 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 author that implementations exists for EAP-SIM except Tseng's proposition [31] 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 mobile that has been current contract with their MNO. It still is responsibility of authenticator to check AuthZ and let only admin mobile access management network.

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 only thing what you can mine from 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 problems on inserting the admin role information in RADIUS message. Operator or proxying RADIUS, on the other hand,

does not necessary know about roles, without BaaS, so there we need link between them to get role information inside RADIUS packet. It seems, that AuthZ data must be mapped during first phase of EAP-SIM AuthN, when IMSI still is available, and in some way forward that map to the proxying RADIUS servers. These issues remains to be solved as our model uses AuthN only as Full-authentication and temporal identities are not used.

4.8 Similarities with Lock-and-Key method

The method is very similar to concept used on routers to dynamically enable access to certain parts of network by first letting the user to log in to the router. Device provider Cisco calls this "Lock-and-Key" access and uses dynamic access list to implement it. [25, p.117]

Difference here is that 802.1X protects access to network. Smartphone does not have any access to network before AA, while in Lock-and-key access is already open to network and successful login to router opens access to another segments through it. Both methods can have RADIUS as a authentication server.

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

4.9 Chosen solution

[wrap up of solution]

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

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

Above it was mentioned, that the local controller delivers changes to each device. On this work, it is 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. Furthermore, the authenticator is presented as the access point 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, variation of design is, that not every change needs

to go through the local controller and so process does not always need interaction from the user.

Critical changes are those, where network topology changes so 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 previous can first be seen on division of homenet to guest and private network and extensions for homeworkers instead of office.

5. IMPLEMENTED SOLUTION

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

5.1 EAP-SIM authentication test bed

Used physical devices are AP and laptop. AP is running OpenWRT firmware. Laptop's software are WPA-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 is therefore physically split-brain: It asks from itself for AA. Figure 5.1 shows how EAP-SIM AuthN messages (dashed and solid arrowed lines) flow when using simulated WPA-supplicant and HLR-AuC as simulation environment.

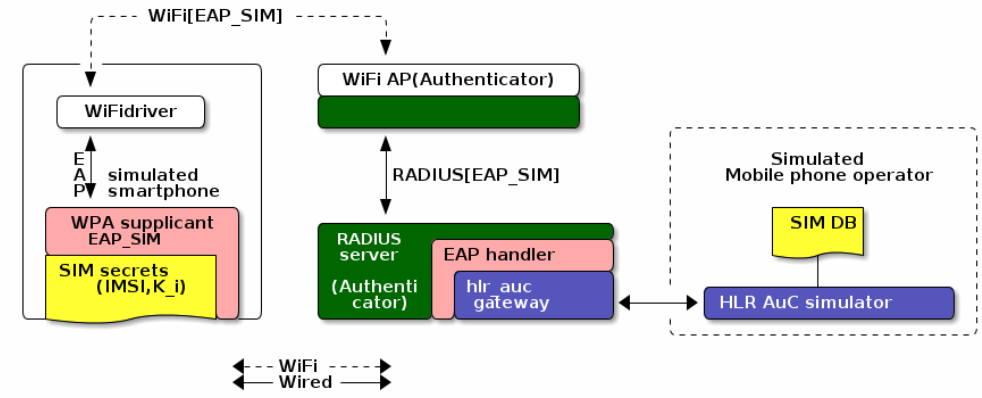

Figure 5.1 *EAP-SIM AuthN messaging in simulation testbed*

Jouni Malinen's software package *HostAP* can be thought as an reference implementation providing all necessary components: WPA-supplicant, Wireless Access point (AP), HLR-gateway (for GSM networks) and EAP-endpoint with or without RADIUS-server. HSS would replaces HLR in 3G/UMTS networks. [21].

For more realistic test, OpenWRT AP is used instead of *hostapd*'s access point 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.

5.2 Detailed description of test runs

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

First tests did not go as planned. There was no indication of SIM method present in captures, only indication of security was "Open access". Nokia e90 Symbian, with 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 for that specific card.

In this point, physical phones were put aside and simulated SIM-card was used. After WPA-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 needed programs. It also recorded used configurations, logs, and traffic captures for later analysis.

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

Terminating 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 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 [10] to 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 of RADIUS UDP port 1812.

[Following AWAY. left from early phases]

Time limited access can perhaps made with session-timeout parameter in ACCESS-ACCEPT (or ACCESS-CHALLENGE) packet using type field = "29". This parameter tells the authenticator how many second maximal the supplicant can have service.

[This cannot be type field 29!] More specifically, what action authenticator should do after termination becomes. It has values of either 0 (default) or 1 (radius request), which would mean that authenticator may send new ACCESS-REQUEST to RADIUS server.

But that would eliminate direct authenticate-only RADIUS cases [*were there any? I don't remember what I meant by this. Maybe that we needed only to have authentication for access which in turn enables modifications*] Is it then that with value 0, authenticator does not send ACCESS-REQUEST to RADIUS server, but client still can automatically send it without user's acceptance?

- forced logout, like in captive portals, where RADIUS is not used.
- no straightforward solution exists within RADIUS
- AP is programmable with luci, which is used in configuring routers. It also could run some existing WWW-access portal [-> reference to No Internet connectivity link is

[Back in track: this can be left here]

Offline changes includes cases where smartphone is not at home or when there is no internet connection available. If connection to Internet is down, full SIM authentication will not work, because it needs co-operation from Internet, namely

from MNO. Simple solution would be sending one-time password to predefined phone via an SMS, but what entity would then check that? Authenticating server, which has no internet connection should have way to check that one-time password received via SMS is correct.

Solution for this could be co-existing WWW-based authentication, that is web-page where credentials could be entered. Software would run in AP. Existing solutions for this are for example Chillisoft or NoCatAuth. That means open access to the portal site must be provided.

Full authentication uses IMSI, which is the identity of phone's SIM. Fast re-authentication would use temporal identity TMSI, which can change each time the AuthN request had been sent. Mapping is cached on authenticator and round-trip and handling at HLR is so eliminated.

IMSI is 14 or 15 digit long number and presented as a composition of digits belonging to MCC(2 digits), MNC(2-3 digits) and MSIN(10 digits). As for TMSI, it is composed of pseudonym and realm part and can be string. So, one can send `my-string-which-can-change@...operator.domain` instead of IMSI number (or `IMSI@......operator.domain`) as an identity.

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

Wireless capture between WPA-supplicant and AP was made on WPA-supplicant's end-point, before it left wireless card. Capture was not made in monitoring mode, so not all 802.11 details in data packets were captured. Because the focus was not in the radio channel but instead in the EAP messaging, that was not problem. [36].

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

- flow of messages, timing, size, attributes

Even, when authentication would not complete fully, authenticator still receives identification claim from mobile but as there is no AuthN, then there is no proof of identity in that case.

IMSI is sent first time already on second EAP message from wpa-supplicant to AP (see Figure 2.2, message 2.) Same in tests made 150123-155714, source: testit/demot/aps150123-155714/ Capture is from mobile client, when it has received 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 WPA-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.

We often notice in communication protocol designs, that they have not been fully optimized. For example name to IP requests in Domain name system (DNS) encode IPv4 address as 16 byte string instead of 4 byte integers. Here it would not take much space to save IMSI which is 15 digits at most. As $2^{50} = 1125899906842624$, then 50

is smallest amount of bits needed to encode smallest, 16 digit long decimal in binary. 15 digits therefore can be encoded with 50 bits, and it takes no more than seven bytes (not 15) to save that information as can be seen in equation 5.1. In reality, such optimization, while saves some bytes, merely leads to misunderstandings.

$$\left\lceil \frac{50}{8} \right\rceil \frac{bits}{bits/bytes} = \lceil 6.25 \rceil \, bytes = 7 \, bytes \tag{5.1}$$

[TBD: Check other examples, where they are not the same!]

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 WPA-supplicant both in identity and cred 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 to MNO service provider producing AuthN [while AuthZ is already taken care of with cloud service contract.] [TBD, leave cloud out] For AP, modifications are minimal. Needed settings are WPA mode to WPA-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 WPA-Enterprise), and AuthN method EAP-SIM. Smartphone modifications can exist together with other profiles Different SSID makes that separation possible.

6.2 Estimating time to authenticate EAP-SIM

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 access point 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 LANGATONWPA 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 mobile 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 used A3/A8 algorithm 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 for part that implements AuthN for own operator and for part, that is free to use by some other operator. Instead, 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. Interesting use case have been Chaos Computer Club's international CCC-camps [3], 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, of interest here is only that GSM encryption could be used and SIM-card secrets were known to organizing operator. On the other hand, empty GSM cards for testing can be cost as much as 18 euros a piece (webshop-quote [1]).

6.4 Platform specific issues

For clients, there is no need for public key infrastructure (PKI) unless EAP-PEAP is used. There are smartphones, that do not have EAP-SIM yet available. For example there is no support for Android although EAP-SIM (and -AKA) methods do work Android 4.x and iOS 5.x. [17].

Generally, what is needed to bring EAP-SIM support to open source smartphone is *pcsc-lite* for accessing SIM card, wpa_supplicant for wpa client, and possible used connection manager (*connman* or *wicd*). This is in line, what was done in testing, without pcsc-lite because SIM card was not used but a file.

For platforms, if OpenWRT is used, greatest problem is the size of memory which can be less than 32Mbytes. Base wpa software included in base OpenWRT installation is small, but it does include RADIUS server part nor EAP-SIM handling..

Software has other limitations. Freeradius2 is not included yet in OpenWRT. It would also be based heavily on current Perl environment which itself may be a space

hog. Currently, as of 1.7.2014, there is no support for EAP-AKA on freeradius2 even when there was support on version 1.1.4. [11]. EAP-SIM is supported. 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 described methods of homenet management delegation. Following paragraphs divides them into Confidentiality (privacy), Integrity, and Authenticity. Accessibility is also discussed.

The purpose of message confidentiality in authentication phase is to hide identity and possible delivered secrets from eavesdroppers. EAP-SIM cannot be considered confidential during first message exchanges, but later identity can be hidden.

If EAP-SIM is used as the only EAP without EAP-PEAP, then there are no mitigation for revealing the IMSI on the first message and it leads to privacy issue. 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 network to authenticate itself to the phone in thus GSM allows man in the middle attack device called IMSI-catcher to fake as being a base station. When mobile phone tries to attach to fake base station, it reveals its IMSI number. Further, because the base state is responsible for chosen encryption, it can order phone to not encrypt traffic or to use only weak encryption thus revealing all data, calls, and texts [29]. Mitigation for IMSI-catching would be to disable GSM (2G) usage altogether from phone if that is possible.

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.

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 [24].

6.6 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 [18].

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 [23, Section 2.2.1]. On the other hand, use of an alternate RADIUS server serves better than waiting for TCPs reliable delivery.

6.7 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* [33].

RADIUS uses MD5 hashing and shared secrets. Because of weaknesses of MD5 hashing (MD5Attack [10]), 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. RADIUS messages are protected from basic message injection without previous knowledge by those integrity checks, which use session secrets, hashes and exclusive OR-function (XOR).

In scenario III(Figure 4.3), there was proxying RADIUS between authenticator and MNO. When MNO notifies authenticator that a smart phone has been authenticated, then authenticator (AP, functioning as a RADIUS-client) hooks that message and usually just grants smart phone access to 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.

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

Earlier in RADIUS analysis, prevention of replied messages was mentioned. Reusing 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 same, but TMSI should be different for call and EAP. Haverinen [15] explains how special RAND numbers can be used to differentiate use of 3GPP and LAN contexts.

Re-authentication and termination can bring unexpected results. If SSID changing, which was introduced in mitigation Section(6.9) was in use, fast re-authentication should be forbidden [8, 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 customer: “Mobile” and “Mobile Eapsim”. The latter network did not ask customer for connection and used smart phones SIM automatically. Unfortunately, it also charged users for using Wi-Fi connections without their knowledge. [20]

If one can read and write data through SIM card’s API, he could try to get information (SRES, K_c) by brute-force. Fortunately SRES and K_c are never sent in clear, but inside digested MAC. Additionally SIM card can be programmed to answer only limited amount of challenge request, for example 65535, which in normal usage would be enough, but in brute-force challenges would soon fill up.

6.9 Mitigation methods

To mitigate risks 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 management network is needed only then when changes are challenged. Why then not just enable management radio network then? Then there were less networks for users to choose from. Enabling management network could be programmed through OpenWRT router’s lUCI-interface but preliminary tests showed, that it also disconnects existing Wi-Fi connections and may even restart AP, which certainly would not be wanted. Some other methods needs to be invented to avoid denial-of service.

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 exactly the target SSID name. 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 then again that would defeat the purpose of easy access.

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 would be the only security method. Disabling or hiding merely gives one security layer more so it is not real security method.

When the usage here is to always renew SSID name then hiding actually could add security. AP does not reveal any information before the client has tried to connect to it, so at least the time window for attacks is minimized. Indeed hiding can have privacy enhancing effects: Lindqvist [19] uses hidden APs to protect privacy of clients who actively probe for visited APs and would otherwise reveal their visited places.

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. repeaters) are in use. Proximity so brings a minimal extra layer for preventing attacks just like network cloaking as 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.

6.10 Discussion

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 of 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.

7. CONCLUSION

Homenet's future needs in configuration management have been described. As an example, change of authority and delegation of control to third parties are needs that have been presented. A method to approve changes indirectly has been proposed. The approval follows from successful authentication and authorization with EAP-SIM method by mobile phone.

Complexity of existing models in interworking was one motivator for the work. Research work on subject did reveal some of the reasons for the complexity, that are difficult to overcome with simplistic methods without in the same time losing some of security.

As results, a real working EAP-SIM test bed with fake credentials and fake mobile operator representing EAP-SIM authentication flow has been shown. A dual-role model, which binds smartphone to homenet and grants right to make changes has been proposed. Working with management network is indirect way to approve changes. [or: An indirect way to approve changes is achieved by binding authorized access to management network.]

There are some obvious weakness in proposed solution. Possible usage must carefully check the safety limits even when RADIUS-protocol still has strengths in security today. Thesis only scratches bootstrapping problems and work on bootstrapping the homenet needs to be studied more thoroughly. One could use tickets in Kerberized way as in GBA. Software implementation as app is needed to the smartphone.

With 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. Studying HS2.0 few steps further would bring mobile phone internet off-loading on Wi-Fi networks and that would be the missing link in interworking between two worlds.

BIBLIOGRAPHY

[1] [Online]. Available: http://www.smartjac.biz/webstore/samples-to-order/smartjac-test-sim-configurable-options

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

[3] "Gsm - 28c3 public wiki," accessed: 2015-05-10. [Online]. Available: https://events.ccc.de/congress/2011/wiki/GSM

[4] 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

[5] 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

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

[7] 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

[8] 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

[9] 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

[10] 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

[11] 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

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

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

[14] 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

[15] 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.

[16] 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

[17] 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

[18] 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.

[19] 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

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

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

[22] 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

[23] 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

[24] 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

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

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

[27] 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.

[28] 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

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

[30] "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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B978159749231750006X

[31] 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

[32] 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

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

[34] 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

[35] 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.

[36] Wireshark community, "Capture setup in Wireshark," accessed: 2015-04-20. [Online]. Available: https://wiki.wireshark.org/CaptureSetup/WLAN

[37] 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

A.1 shell, logging options

```
#!/bin/sh -x
# Shell to start programs needed to demonstrate EAP-SIM authentication
# 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 -i 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 creds

[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:cb9cccc4b9258e6dca4760379fb82581"
}
```

```
cred={
        imsi="1232010000000000"
        milenage="90dca4eda45b53cf0f12d7c9c3bc6a89:cb9cccc4b9258e6dca4760379fb82581"
}
```

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