TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Seppo Heikkinen

Applicability of Host Identities in Securing Network Attachment and Ensuring Service Accountability

Julkaisu 1004 • Publication 1004

Tampere 2011

Tampereen teknillinen yliopisto. Julkaisu 1004
Tampere University of Technology. Publication 1004

Seppo Heikkinen

Applicability of Host Identities in Securing Network Attachment and Ensuring Service Accountability

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 25th of November 2011, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2011

ISBN 978-952-15-2684-8 (printed)
ISBN 978-952-15-2711-1 (PDF)
ISSN 1459-2045

ABSTRACT

IP is often seen as the “lingua franca” of modern communication. It provides interoperability across various heterogeneous link layer technologies and enables access to a rich set of services available in the Internet. As the number of users has exploded, it has been essential to ensure that the establishment of connectivity is a relatively painless procedure for the ordinary users. However, often the ease of use overcomes the requirement to have proper security in place. This is evident in the current practise of configuring the IP access. With the proliferation of ubiquitous and wireless access, such security concerns become more profound.

IP access is the part of the attachment procedure that a user has to go through in order to enjoy interworking services. Thus, the attachment dictates the steps that need to be taken in order to enable communication between two entities, which often comprise of the user device and the access point capable of providing interworking services. This thesis investigates mechanisms to ensure the security of that attachment procedure. The approach taken leans heavily on the ideas presented in the development of Host Identity Protocol (HIP), a current Internet Engineering Task Force (IETF) experimental standard. Thus, as a baseline, the nodes are expected to be in possession of secure identities, which can be bound to the configuration procedure in order to enhance the security properties of the attachment process. In essence, these identities are names for which the nodes are able to provide a proof of possession without having to resort to external parties. However, the external parties, for example, trusted third parties, can still be used to enhance liability, that is, ensure that a known entity will ultimately cover the generated costs.

In order to ensure liability, one needs to find an assured way to account for the actions taken, especially if the accounting is used as a basis of compensation, which most often involves payment. Such assured accounting is another focal point of this thesis. The thesis describes how host identities can be employed to produce non-repudiable evidence in a typical IP-based service. In addition, the scheme allows devising a solution, which takes into account the granularity of the service provisioning. In other words, the participants of the provisioning are able to control their level of commitments, so that neither party is able to get an unfair upper hand. Thus, if no service is provided, provision of undeniable usage evidence can be terminated. Similarly, if no user-committed evidence of service usage is provided, service provisioning can be terminated.

The above points are also considered from the point of view of service provisioning platforms, mainly the IP Multimedia Subsystem (IMS). Such systems have been kept tightly in control of one entity, such as an incumbent operator. The concepts of secure identities and non-repudiable evidence are used to enhance the system, so that more technical incentives for moving away from the strict single administrative domain concept can be provided. This is especially beneficial in a future networking environment, where the interactions between the operator level entities become more dynamic in nature.

While this thesis considers the technical functionalities to enhance the security properties of the evolved networks, there are various hindrances to such visions. The deployability of new solutions, such as HIP, is challenging to well-established platforms, if the migration cannot be done in a compatible way. Also, financial motivations, especially those of dominant entities, do not always favour more open approaches.

PREFACE

The research in this work has been mainly carried out at the Department of Communications Engineering at the Tampere University of Technology during the years 2006-2011. However, the origins of this work trace back to 2004-2005, when the author was working for Elisa in a partly EU funded project, Ambient Networks, in which the selection of securing configuration provisioning and network attachment as a research topic was more by accident than the result of a carefully laid plan. This led to many interesting and fruitful discussions with the people of the security work package, for which I am grateful.

This work has been funded by the Graduate School in Electronics, Telecommunications and Automation (GETA), a graduate school where I had the privilege to be during 2006-2010. Additionally, this work has been supported by TEKES through the Future Internet ICT SHOK programme, phases 2 and 3.

I would like to express my gratitude to my pre-examiners, professor Tuomas Aura and professor Josef Noll for their insightful remarks and well-founded criticism, which has helped to improve the quality of this thesis. I would also like to thank professor Josef Noll and professor Andrei Gurtov for agreeing to act as opponents in my dissertation defence. My thanks also go to my supervisor professor Jarmo Harju for giving me a place to work during all these years at the department. Additionally, I would like to extend my thanks to my co-authors as well as to my colleagues here at the networks and protocols group and all the administrative people who have aided me in my everyday work. Also, my thanks go to Michelle Alexander for her valuable help in proof-reading this thesis. Lastly, I would like to extend my warmest gratitude to my parents and to my sister and her family for all their support they have given me over all these years.

Tampere, November 2011

Seppo Heikkinen

TABLE OF CONTENTS

LIST OF PUBLICATIONS

This thesis contains an introductory part and seven publications. In the text, the publications are referred as [P1],[P2],...,and [P7].

[P1] Heikkinen S., Tschofenig H., "HIP Based Approach for Configuration Provisioning", in *Proceedings of The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, Helsinki, Finland, Sep 2006.

[P2] Heikkinen S., "Authorising HIP enabled communication", in *Proceedings of The 10th International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS07)*, San Diego, USA, Jul 2007.

[P3] Heikkinen S., "Applicability of Host Identities to Implement Non-Repudiable Service Usage", in *International Journal On Advances in Systems and Measurements*, vol. 1, nr. 1, pp. 14-28, 2008.

[P4] Heikkinen S., Siltala S., "Service Usage Accounting", in *IEEE Vehicular Technology Magazine*, vol. 6, iss. 1, pp. 60-67, 2011.

[P5] Heikkinen S., "Security and Accounting Enhancements for Roaming in IMS", in *Proceedings of The 6th International Conference on Wired / Wireless Internet Communications (WWIC08), Lecture Notes in Computer Science 5031*, Tampere, Finland, May 2008.

[P6] Heikkinen S., "Establishing a Secure Peer Identity Association Using IMS Architecture", in *Proceedings of The Third International Conference on Internet Monitoring and Protection (ICIMP08)*, Bucharest, Romania, Jul 2008.

[P7] Heikkinen S., Silverajan B., "An Architecture to Facilitate Membership and Service Management in Trusted Communities", in *Proceedings of The International Conference on Computational Aspects of Social Networks (CASoN 2009)*, Fontainebleau, France, Jun 2009.

LIST OF ABBREVIATIONS

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorisation and Accounting
AIP	Accountable IP
AKA	Authentication and Key Agreement
ANAP	Ambient Network Attachment Protocol
ARP	Address Resolution Protocol
AS	Application Server
BEX	Base Exchange
BGCF	Breakout Gateway Control Function
CA	Certificate Authority
CGA	Cryptographically Generated Address
COPS	Common Open Policy Service
CSCF	Call Session Control Function
DAD	Duplicate Address Detection
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DN	Distinguished Name
DNS	Domain Name System
DoS	Denial of Service
EAP	Extensible Authentication Protocol
EAPOL	EAP Over LAN
EP	Enforcement Point

ESP	Encapsulating Security Payload
GBA	Generic Bootstrapping Architecture
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
HBA	Hash Based Address
HI	Host Identifier
HIP	Host Identity Protocol
HIT	Host Identity Tag
HMAC	Hashed Message Authentication Code
HSS	Home Subscriber Server
IBC	Identity Based Cryptography
IBCF	Interconnection Border Control Function
ICMP	Internet Control Message Protocol
I-CSCF	Interrogating CSCF
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPsec	IP Security
ISIM	IMS Subscriber Identity Module
ISP	Internet Service Provider
LTE	Long Term Evolution
MAC	Media Access Control
MCGA	Multi-key CGA
MIC	Message Integrity Code
MIKEY	Multimedia Internet Keying
MitM	Man in the Middle

NAI	Network Access Identifier
NAT	Network Address Translator
ND	Neighbor Discovery
NRD	Non-Repudiation of Delivery
NRO	Non-Repudiation of Origin
NRR	Non-Repudiation of Receipt
NRS	Non-Repudiation of Submission
P2PSIP	Peer to Peer SIP
PAA	PANA Authentication Agent
PaC	PANA Client
PANA	Protocol for carrying Authentication for Network Access
P-CSCF	Proxy CSCF
PDP	Packet Data Protocol
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PLA	Packet Level Authentication
PSIRP	Publish-Subscribe Internet Routing Paradigm
QoS	Quality of Service
RADIUS	Remote Authentication Dial In User Service
RSN	Robust Security Network
SA	Security Association
SAML	Security Assertion Markup Language
S-CSCF	Serving CSCF
SDP	Session Description Protocol
SEND	Secure Neighbor Discovery
SGSN	Serving GPRS Support Node
SIP	Session Initiation Protocol

S/MIME	Secure/Multipurpose Internet Mail Extensions
SNMP	Simple Network Management Protocol
SOA	Service Oriented Architecture
SPKI	Simple Public Key Infrastructure
SSH	Secure Shell
SSID	Service Set Identifier
SSL	Secure Socket Layer
TLS	Transport Layer Security
TTP	Trusted Third Party
UICC	Universal Integrated Circuit Card
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wireless Protected Access

1. INTRODUCTION

The world of telecommunications is changing. The traditional paradigms have slowly given way to the ideas which promote the dominance of IP (Internet Protocol) as the medium for exchanging information among different parties. It is true that the heterogeneous networks require some form of common layer, which enables them to interwork seamlessly and give the user an image of flawless communication available everywhere, independent of time and location. While this common layer indeed is able to transport units of information, there is still the question of how to establish this communication and what procedures are needed to make sure that the information flow takes place as intended. Even more importantly, we need to know whether we really are conversing with the intended entity.

In this work, we discuss the preliminaries that have to be performed before communication can take place securely. In other words, we present technologies and security enhancements for the network attachment procedure, which enables two different entities to come into contact and start conversing with each other and exchange information. This can also be called the bootstrapping of communication. On the other hand, this communication usually has some goal; in other words, it strives to fulfil some service need. Thus, even though the communication might start, it is another thing as to whether the service need is satisfied. This is especially a concern if the other party has committed to something like payment in the course of bootstrapping but then fails to receive anything in return. This calls for measures to ensure the accountability of actions, i.e., valid proofs that someone was involved in a certain action or event.

Currently, such proofs are generally based on information stored in the backend systems or logs, which is then fed to the billing systems. This already shows the asymmetry of the situation, that is, the user is left without any control, but it is further emphasised by the fact that this proof is basically just a textual log entry with no strong binding to the entities involved. For example, if given to a neutral third party for evaluation, there would be no real proof that the entry actually relates to the specified entity.

So, the foremost concern of this work is the security of these measures, but one should not forget issues like performance and usability. It is, after all, somewhat easy to make very secure systems, but the performance and usability of such systems is most likely quite questionable. From this it follows that people are not going to use such systems,

especially if the effort needed to setup the system is increased because of the employed security measures. The other side of the coin is whether there is incentive for the providing party to implement such a system. Usually this means that there ought to be an economic benefit for doing so.

1.1 Objective and scope of research

The objective of this thesis is to present and analyse mechanisms which can be used to increase the security of the network attachment and the subsequent service provisioning. In essence, this considers the mechanisms based on the existence of host identities, which will provide enhanced accountability as compared to the current situation. Thus, it is assumed that the entities have secure names in the form of cryptographical identifiers, or crypto-ids. With such names, it is possible to provide proof of possession of a certain identifier and refer to the entities outside mutual exchange. This allows devising always-on security solutions where there is always a default baseline level of security, but it can be upgraded according to the specific requirements of different application scenarios, e.g., with the help of trusted third parties. Additionally, this thesis devises a non-repudiable mechanism for service usage.

The scope of host identities is originally intended to be on the network level, but this work also considers their extensibility to other layers. This kind of cross-layered approach allows the procedures on different layers to benefit from the same infrastructures. In other words, the procedures executed on the network layer can also be used to provide security for the actions taken on higher layers. As an example of this kind of extension, applicability of host identities in the context of IP Multimedia Subsystem (IMS) is analysed. Additionally, this is considered from an evolved network perspective, i.e., ambient networking, in which entities exhibit more dynamic interaction in their relationships, for example, roaming agreements.

1.2 Research contribution

This thesis analyses the applicability of host identities to enhance the security properties of attachment scenarios. The first case is the typical network attachment and the ensuing configuration procedures, which can be made more secure with the application of host identities through the protocol suggested in this work. This can be just rudimentary security built upon the sameness property of the entities, suited for cases where preconfiguration is not feasible, or it can be further enhanced with authorisation statements, which can be integrated to the attachment procedure and give more credibility to the configuration requests or the validity of the configuration provisioning.

The second case relates to the service provisioning and the related accountability properties. This includes devising a protocol with the non-repudiation property, which states that the parties are not able to deny their involvement in the communication. This is

further enhanced with a fairness mechanism, which is used to control the risk the parties involved in a service transaction take. In other words, the service user might be committed to provide compensation for the usage of resources, but could face the danger of not receiving the promised service. From the service provider perspective, there might be the danger that a user might later deny involvement in the service, thus avoiding payment. The feasibility of the aforementioned concepts are also analysed within the context of service management platforms, such as IMS. In addition, this thesis suggests extending them to the formation of trusted communities by devising a conceptual architecture.

1.3 Outline of the thesis

This thesis is comprised of an introductory part and the accompanying research publications. The introductory part provides background information about the described cases and sets the context for the publication to which they are related. The publications can be grouped under three themes, which relate to initial network attachment procedures (P1, P2), non-repudiation mechanisms (P3,P4), and the applicability of these ideas in the context of the service management platform (P5,P6,P7).

The second chapter of the introduction discusses the security fundamentals, that is, the threats and the security objectives considered in this thesis. The third chapter gives an overview of network attachment and how the identity based approach can be used to answer to the security shortcomings of the portrayed mechanisms. The fourth chapter discusses the accountability and non-repudiation aspects giving background information and setting the stage for the provided research publications. The fifth chapter combines the issues presented in the previous chapters and applies these in the context of IMS. The sixth chapter gives the final conclusions before the accompanying publications.

1.4 Summary of publications

The first publication, P1, shows how Host Identity Protocol (HIP) can be used to secure the configuration provisioning of network attachment. HIP protocol messages are used to piggyback the configuration information in a manner, which takes into account the current stateful provisioning mechanisms. The author’s contribution entails the original idea and the design and analysis of the protocol specifics.

Publication P2 discusses the aforementioned HIP in the context of authorisation. It presents and analyses different alternatives for the inclusion of authorisation tokens to the basic HIP, which at that time was still in the draft stage. Special emphasis was given to the practicality aspects, i.e., how well the different mechanisms would be suited to the actual protocol exchange.

Publication P3 presents a system, which is based on HIP and provides it with non-repudiation properties. In other words, it considers HIP as a protocol, which can be used to negotiate the service usage between the participants and the relevant evidence of the service usage can be provided, i.e., accounted through the protocol. It is shown how the properties of HIP are well suited for setting up an identity based framework for service provisioning.

Publication P4 provides an extension to the previous framework by taking into account the existing accounting solution Remote Authentication Dial In User Service (RADIUS) as it is commonly used in WLAN scenarios to authenticate the user. Thus, it provides a natural mechanism for interacting with the existing operator backend systems. The publication also presents the actual implementation of such a system. The author's contribution was the overall design of the system and the needed protocol exchanges.

Publication P5 describes application of the Host Identity Protocol in the context of IMS to provide security and accounting for a roaming user. The publication presents ideas of extending IMS to support dynamic operator relationships in an ambient computing setting.

Publication P6 extends the typical IMS interaction between the end entities by introducing additional security measures. The main point is providing the assurance to the received identity, which can be used to establish a secure identity association between the end entities. Additionally, HIP signalling can be exchanged in-band by taking advantage of the existing IMS infrastructure.

Publication P7 approaches service provisioning from a more abstract and general viewpoint. It defines a conceptual architecture for identity based trusted communities, which can function as trust roots for the members in using and providing services. It is a mixed model, which could support both centralised and distributed settings. The author's contribution includes the co-design of the architecture and the requirement capture of the community identity management.

2. SECURITY FUNDAMENTALS

2.1 Communication threats

Threats are potential violations of security [Bis02]. Attack, on the other hand, is an action, which could cause such a violation. Hence, often those terms are used interchangeably. One categorisation for communication threats is given in [Sta98], which lists interruption, interception, modification and fabrications as types of attacks that can affect the operation of a network. It is worth noting that even though different kinds of classifications are used for actions, they may still lead to the same end result from the point of view of the victim. Thus, threat classification should be seen only as a tool for finding the commonalities between different kinds of threats. This helps in devising security solutions that can take into account several types of attacks.

Interruption means that the message flow is not allowed to reach its destination or that the normal behaviour of the system is prevented. This could happen due to various reasons, such as the adversary jamming the wireless link with its transmission. It is also possible that the adversary intentionally directs some extra traffic to one party of the communication, thus making it use more of its resources to cope with the increased load and possibly resulting in extra costs. When the load gets high enough, the system stops responding to the legitimate traffic. This is generally called denial of service (DoS), which is a serious concern for any modern telecommunication system. An enhanced version of this is called distributed denial of service (DDoS) and it is a result of several different traffic sources sending a large amount of messages either intentionally or unintentionally to one target. Quite often the traffic sources are innocent per se, but have been subverted by an attacker, who is able to control their behaviour. DoS can also be accomplished through means that cause the target to use all of its computational resources. This can happen, for example, due to some design flaw in a cryptographic protocol or asymmetry of devices, i.e., a mobile device usually has less computational resources than a normal desktop computer.

When an unauthorised party gains access to the information, we are dealing with an interception kind of threat. This can happen simply by eavesdropping, that is, listening to the traffic if the messages are sent in clear, or, in the case of encryption, it can involve various sophisticated techniques, such as cryptanalysis and traffic analysis, for compromising the confidentiality of the communication. Sometimes, however, protocol design or implementation has been so flawed that this is relatively easy. The protection

used in the first 802.11 wireless LANs is one notorious example, where the adversary was able to find the secret key by monitoring the protected traffic [Flu01].

Modification is a more serious version of interception as the adversary is also able to modify the content of the messages. These kinds of attacks can have a considerable impact on the overall system, because the integrity of the distributed computing platforms can be compromised if the forged information is used as a basis for deciding which type of actions to take, e.g., allow or deny access. It is, thus, possible to open additional holes for the adversary, who can gain further access to the platform. Modification can also result in DoS, if, for example, dynamic configuration instructions are modified in such a way that the target ends up in a state where no communication is possible, because the installed configuration does not match the network setup. Modification attacks can often be called Man in the Middle (MitM) attacks as the adversary has managed to insert himself into the communication path of the communicating parties.

The fourth type, fabrication, takes place when an unauthorised party is able to insert messages or objects into the system. The difference with the modification attack is that the messages are new and possibly counterfeited to in some way resemble legitimate messages, but they can also be closely crafted messages that are known to cause some undesirable effect. This is often used to take advantage of the implementation deficiencies, such as susceptibility to buffer overflows, which can allow unauthorised access to the system.

2.2 Security objectives

The threats presented in the previous section are just one viewpoint of looking at the security of the system. On the other side of the coin are the security objectives the system is expected to meet that are an important part of the system design. They can be seen as high level requirements for the operation of the system, or another viewpoint would be the security services the system is expected to provide. Such requirements are also needed in guiding the design of solutions presented in this work. Quite often in information security confidentiality, integrity, and availability (CIA) are mentioned [Bis02], but we adopt here the grouping given in [Sel05], which is availability, authorisation, accountability, and assurance. To a great extent, it follows the grouping given in [NIST01].

Availability is intended to ensure that the system works promptly and service is not denied to authorised users. This clearly relates to the DoS concerns given earlier and is tightly linked with the interruption; that is, interruption attacks generally affect availability. Note that this also includes prevention of unauthorised deletion of data, be it either through intentional or unintentional means.

Authorisation relates to the actions that are permissible only to the intended users. This can be seen to include both integrity and confidentiality objectives as well, because violation of these happens through unauthorised action (in fact, [NIST01] lists the previous objectives instead of authorisation). In other words, no unauthorised entity is able to alter the system or data nor is able to access the information. [Sel05] also categorises privacy under authorisation, but the concept of privacy is likely to go beyond this as it includes both information about the user as well as the ways the user interacts with the environment. The need for privacy is also controlled by legislation [EC0258]. More discussion on privacy can be found, for example, in [LAPS03].

Accountability is the requirement that the actions of an entity may be traced uniquely to that entity [NIST01]. This encompasses authentication, auditing, and non-repudiation. In this sense the authentication means verification of the claimed identity, and non-repudiation relates to the fact that the entity cannot deny its involvement in an activity. To some degree it is possible to include access control under accountability, but it also is affected by authorisation, because generally, access control is responsible for checking that an entity is authorised for access and not so much the authenticity (even though access control may also be interested in who uses the resources). This decoupling of authorisation and authentication is discussed later.

Assurance dictates that the previous objectives are met [NIST01]. It is the basis for confidence that the security measures, both technical and operational, work as intended to protect the system and the information it processes. In other words, it relates to the actual implementation and its correctness, so that it has the required functionality and has sufficient protection against unintentional errors and intentional penetration. One could also state it as a basis of how much one can trust the system [Bis02]. From this perspective, malicious activity by authorised users, for example, an administrator, could fall into this category. Assurance is also an important part of the Common Criteria for evaluating systems, for example [CCA09].

2.3 Trust relationships and key validity

Often in the system design process it is stated that certain entities have to trust each other in order for the system to be secure. However, almost as often, this is not further clarified or stated that it is established through some out of band means. This kind of treatment of security is not uncommon and is almost equivalent to just stating in the design requirements that the system has to be secure. Trust could also be approached from the perspective of above mentioned objectives: one believes that the system will meet the given requirements [Bis02]. However, here we briefly discuss the aspects of trust from the perspective of entity interaction as it falls under our scope in terms of identity validity (or key validity, as it is the basis of this work). In other words, we use the term trust in a relaxed manner, when we are actually referring to the ability to verify the authenticity of the message sender and the way the key distribution is done. It can be

argued though that trust enters the picture only after the participants start using the exchanged authentic information to change their state, i.e., make decisions. Thus, it is worth noting that trust is a complex matter and cannot be fully discussed in the context of this work (for additional discussion, see, for example, [Gra00]).

It is possible that the entities know each other explicitly, that is, there exists direct trust between the parties. They have learnt through some means, like manual configuration, that the other party is a valid communication partner, e.g., keys of the participants are known in advance. This is not usually a very scalable way and is more suited to small environments like homes. A shared secret and proof of knowing this secret is also enough for direct trust in some systems.

Another way of establishing trust is by opportunistic means. This relates to the leap of faith kind of scenarios, where there is no actual information about the other party, but it is sufficient to know that the other party does not change during the communication. From the technical perspective, this is sometimes referred to as key continuity [Gut04]. This generally opens a possibility for the MitM attacks, but it requires an active attacker, who tries to participate in the protocol run in the beginning of the communication. In other words, this is more suitable for the cases where there are only passive attackers, such as eavesdroppers. Secure SHell (SSH) is a popular example of this approach [Ylo96].

The parties could take advantage of a third party to establish relationship between them. In a sense, this could be seen as a recommendation as well [Abd00]. This kind of indirect or brokered approach requires that both parties trust this external entity. This can come through, e.g., preconfiguration, which is a common approach in current Internet browser security. This trusted third party (TTP) can take part in the online negotiation between the parties and ascertain the validity of them, at least from its own viewpoint. TTP can also grant assertions, which can be presented to the other party in an offline fashion. These assertions can make statements about the holder. It could, for example, state that this particular entity is authorised to get connectivity service for a certain time period. It is, of course, up to the receiver and the agreement made with the TTP whether such authorisation is relevant for it. After all, the receiver might not trust all kinds of authorisation decisions made by the TTP. As a side note, these hierarchical systems have developed into Public Key Infrastructures (PKI) with the addition of various management processes. They have received criticism, however, due to untransparency of the system [Ell00] (as well as counter-criticism [Ada04]), which, based on recent cases of compromised TTPs, is not completely unfounded [Hal11].

Authorisation can also happen between the communicating parties, if they decide that there is enough incentive to, for example, delegate some actions to be made by the other party on their behalf. Basically this means taking advantage of the trust relationships of the other party, i.e., trust delegation. This could be, for instance, enhancing the effec-

tiveness of signalling by transferring the signalling from the mobile node to the access network. Delegation can be beneficial in traversing heterogeneous environments, when authentications and authorisations made in one domain can be translated to the other domains as well, provided a trusted party exists that can function in both domains.

A variation to the brokered model is given by the web of trust concept. In that, a party can make explicit statements about the validity of another key, but also introduce uncertainty into the statement. In other words, in order to be considered valid, there has to be multiples of such statements from different trusted partners. Thus, instead of forming hierarchies, the web of trust creates meshes. This approach, which could be termed cumulative trust, has been made popular by Pretty Good Privacy (PGP) [Zim09]. In a sense, this could be seen as related to reputation systems, even though PGP does not really consider the history of behaviour.

2.4 Authentication and authorisation

Authentication and authorisation are tightly related to the trust issues presented in the previous section. Authorisation cannot be meaningfully handled, especially without having trust relationships. Authentication on the other hand relates more to the identification and verification of the identity claims. In other words, authentication consists of those two steps: presenting an identifier as identification and presenting authentication information that corroborates the binding between the entity and the identifier [Shi07]. The authentication information used in the verification of claims can be referred to as credentials and is often categorised as something known, something embodied or something held [Amo94]. In telecommunication, authentication is needed to make sure that the messages originate from, and are targeted to, the legitimate party. Additionally, authentication enables traceability (accountability), so that if something has gone awry, it is possible to find the one responsible, or when dealing with compensation it is possible to find the one, who is supposed to pay (liability), even though this also relates to authorisation, that is, who authorised the action.

Many of the current authentication solutions do not expect there to be mutual authentication, i.e., only one party is authenticated. For instance, protected web sites in the Internet employ Transport Layer Security (TLS) or Secure Socket Layer (SSL) and they basically rely on the model where only the server is authenticated (but user authentication can still take place on the application layer). That approach has been quite successful in many typical Internet application level scenarios, but the situation can become more challenging when payment for connectivity enters into the picture. For instance, without mutual authentication a MitM could place himself between the client and the access point and make the client pay for his traffic too. This mutual authentication requirement can be relaxed, if the proper authorisation is presented; this is discussed in the following paragraph. Also, with regard to entity authentication, it can be enough that

the relevant identifiers are authenticated and not the actual entity identities, that is, the real world identity.

Authorisation, as mentioned earlier, relates to the actions, which are permissible by the users. Alternatively, one could talk about access to specific resources. In a telecommunication system one ought to be able to link a relevant authorisation to all the actions, whereas the common approach seems to be that when the user is authenticated, it also implies authorisation as well. In other words, once authentication is done, there is no granularity to the available actions for the specific entity. However, it is often true that authentication is tightly linked with authorisation, when, for example, a policy dictates that certain users are allowed to do certain actions. This naturally requires that the users are first authenticated before the policy decisions can be applied. This does not, however, remove the need to have the possibility of handling those two concepts separately. In some cases it is beneficial to be able to just allow certain entities to perform actions, if they possess the necessary authorisation, without having to authenticate their real world identity. This way the privacy of the entities is better served. A simple example of this kind of mechanism is an authorisation token, which is discussed in [P2]. In terms of analogies, one could consider a movie ticket, which is checked at the door for authorised entry with no ties to the real identity of the user.

2.5 Importance of identifiers

An actor in the system, an entity, can be seen as possessing an identity, which makes it recognizable from all the other entities. An identity can be seen as a collective aspect of a set of characteristics [Shi07] and in order to be able to point to it, you need a name for it. In essence, the identity is represented by an identifier. What is important in our discussion and one of the key points of this work is whether you are able to make the reference securely, that is, only the legitimate entity can claim ownership of its own identifier, and no identifier spoofing is possible. The identifier, on the other hand, can be resolved to the characteristics or names in other namespaces, and you have to be able to have assurance that the binding is authentic. Another viewpoint regarding identifiers is that when all the entities have names it is easier to refer to them in cases where they do not directly participate in the actual conversation, i.e., one could say that the entities should be first class citizens [Ero04]. Many designs lack this kind of property. For example, a client and a home network often cannot name the used access network in a secure and meaningful way. This is even more prominent in the case of middleboxes, which are intermediate devices that can participate in the communication to provide additional functionality, for instance, in the form of address translation or firewalling.

Access decisions are often based on the identifiers. For instance, a policy may dictate that only a certain set of identifiers is allowed to access a resource available at the host. Typically, the presenter of the identifier also has to present some additional information, credentials, which verify that the entity is really the authentic holder of the identifier.

This is not always the case. For example, a WLAN network might require the administrator to create Media Access Control (MAC) lists, which contain MAC addresses that are allowed to attach to the access points, or switches could bind their ports to an observed or configured MAC. Given the ease of changing the MAC of an interface this does little to stop a determined attacker. Similarly, the Service Set Identifier (SSID), which is used to identify a WLAN network, does not give much guarantee for the client that the access point advertising a certain SSID, like Wlan.BigOperator.com, is actually part of an access network you think it is. In other words, you cannot base any authentication or authorisation decisions on that knowledge alone. This is also evident in modern cellular networks, where the client does not have guarantees about the identity of the access network, even though the access network may be in possession of authentication material provided by the home network and able to execute the authentication procedure with the client.

It is obvious that one needs mechanisms to ensure that the used identifiers are legitimate and their holders are authorised to use them. A well known identifier-credential combination for users is login-password, but in order to verify the binding between them one always needs the relevant database that holds that information, and it is easy to misuse that information in various ways because passwords can be snooped or guessed. Thus, it is more convenient and efficient to have methods that allow the other party to verify the ownership without need of any external parties and have assurance that the ownership proof is not easily compromised. Such requirements can be fulfilled by cryptographic identifiers. These identifiers exhibit mathematical properties, which ensure that the identifier-credentials pair is not easily compromised and the verification can be performed using mathematical computations without the involvement of external parties. For example, a public key pair forms this kind of cryptographic identifier. The public key can be seen as the identifier and the private key as the accompanying credentials, although it actually relates to the ability to perform computations with the private key, as the private key is never disclosed to others. Mathematical equations binding the public and private keys are such that it is computationally unfeasible to try to discover the other key without having all the necessary information. In practise this means that it is not easy to calculate the private key even if the public key is known. So, when an entity is using a cryptographic identifier, it is always able to present proof of ownership, which ensures that the presenting entity has in its possession the private key needed to calculate the proof.

There are also cryptosystems, i.e., identity based cryptography, that enable the users to utilise human readable information, like email addresses, as public keys [Sha85][Bon01]. While this may be attractive from the usability perspective, it always requires the presence of a third party, which provides a corresponding private key for the user. As there is a third party in possession of the private key, a degree of doubt is introduced from the point of non-repudiation. A conventional PKI system could be argued to exhibit the same problem in the case of a malicious TTP, but the difference is

that the third party has to create a false certificate for a new key instead of using an already existing key; that is, the earlier agreements made with that key are not compromised. Improvements for the identity based cryptosystems have been suggested, though, so that only the partial private key is generated by the third party [AlR03].

Another important point about identifiers is that once you have the means to securely identify the entities, you can make statements about their relationships and delegate privileges among them. This can be used to extend the trust relationships beyond the mutual interaction without actually having to know to which identity the identifier currently points to. In other words, if a TTP makes a validity statement about an identifier stating that the identifier is authorised to perform a certain action, then all the entities considering the third party to be the authority over that specific action can also consider the identifier to be valid for that context (depending on their individual policies, of course). An additional issue is accountability. One needs valid identifiers in order to be able to be sure that the costs that the entity generates can be charged. Not necessarily from the holder of the identifier, but at least from the entity who was willing to authorise the action and in doing so was willing to accept the liability of consequences of that action.

One further issue relates to the lifetime of the identifiers. Long lived identifiers are beneficial in policy databases and authentication statements, like in typical certificates, because the effort of storing and creating is mitigated by the fact that the information is used over and over again. It is also easier to base long lived business relationships, like subscription, on such identifiers. However, they have the downside of enabling privacy violation, i.e. it is possible to track the movements and actions of a certain individual. Therefore, it is sometimes useful to have the additional ability of using ephemeral or short lived identifiers, which can be discarded after use or some short amount of time. It may not be feasible to add them to databases or access lists or bind full-fledged identity certificates to them, but individual authorisation statements can be made regarding them. Thus, the validity of the identifier comes through the authorisation alone. With short lived identifiers one gains an additional risk management mechanism, which is easier to deploy than, for instance, a revocation infrastructure, which would be needed to manage the possible compromise of long term identifiers. Additionally, a recent PKI compromise with Comodo [Hal11] would suggest that dynamic revocation mechanisms are not extensively used as the browser vendors were quick to provide updates to their software instead.

2.6 Host identity

While not exactly being a fundamental security primitive, here we go briefly through the basics of Host Identity Protocol (HIP) [Gur08] as it, and especially the identity based approach suggested by its architecture, is the fundamental building block of the proposals suggested in this work. In the previous sections we have discussed both iden-

tities and identifiers, but in this discussion the use of those terms is mixed, that is, the host identity may not tell much about the real world identity.

Host identity is intended to create a new namespace, which consists of host identifiers (HI) [Mos06]. Host identity refers to the abstract entity that is identified, whereas a HI is the concrete bit pattern used in the identification process [Mos06]. In essence, a HI is the public key of a signature key pair. An additional concept is the Host Identity Tag (HIT), which is a concise 128-bit representation of a HI and it has been created through hashing. This HIT has the format of an IPv6 address, albeit a non-routable one in the network sense [Nik07]. Conceptually host identities are seen to be between the network and transport layers. Thus, transport layer protocols can bind to host identities (or HITs, rather) instead of network layer entities, such as IP addresses. This allows decoupling the dual role problem of IP addresses, i.e., they are currently acting both as locators and end point identifiers [Mos06]. This better serves the mobility and multihoming requirements of modern networks, because the underlying IP address can now change without affecting the transport layer.

Host identities can use HIP to establish an association between each other [Mos08]. This takes place with a process called HIP Base Exchange (BEX), which comprises four messages depicted in Figure 2-1. The process allows authentication of peers and establishment of a secure state between participants. It also makes an effort to mitigate DoS concerns. In the course of handshake messages, the parties are able to construct keying material, which can be used to protect any subsequent communication, for example, with the help of IPsec Encapsulating Security Payload (ESP) [Jok08]. HIP uses parameter structures to encode the various functionalities to the protocol, hence providing an extensible mechanism for introducing additional functions. Such extensions include, for instance, Network Address Translator (NAT) traversal [Kom10], rendezvous mechanism [Lag08], and mobility [Nik08]. There have also been proposals for lightweight alternatives for BEX, like in [Hee07] and [Mos11].

BEX begins with an I1 message, which the initiating party, called the Initiator, sends as a trigger to the responding party, called the Responder (see Figure 2-1). This message contains the HIT of the Initiator and can also include the HIT of the Responder if it is known. The response to it, R1, contains a puzzle, Diffie-Hellman parameters of the Responder, suggestion of the cryptographic algorithms, and the host identity of the Responder. The puzzle is used as a DoS mitigation mechanism and challenges the Initiator to prove its sincerity with regard to connection setup as it is expected to use computational resources to solve the puzzle. The Responder also includes a signature, which is calculated using the private key of the Responder and does not include mutable parts of the message, so that the signature can be calculated beforehand, thus further mitigating DoS possibility. The Initiator is able to validate the signature and the Responder HIT using the public key given with the host identity.

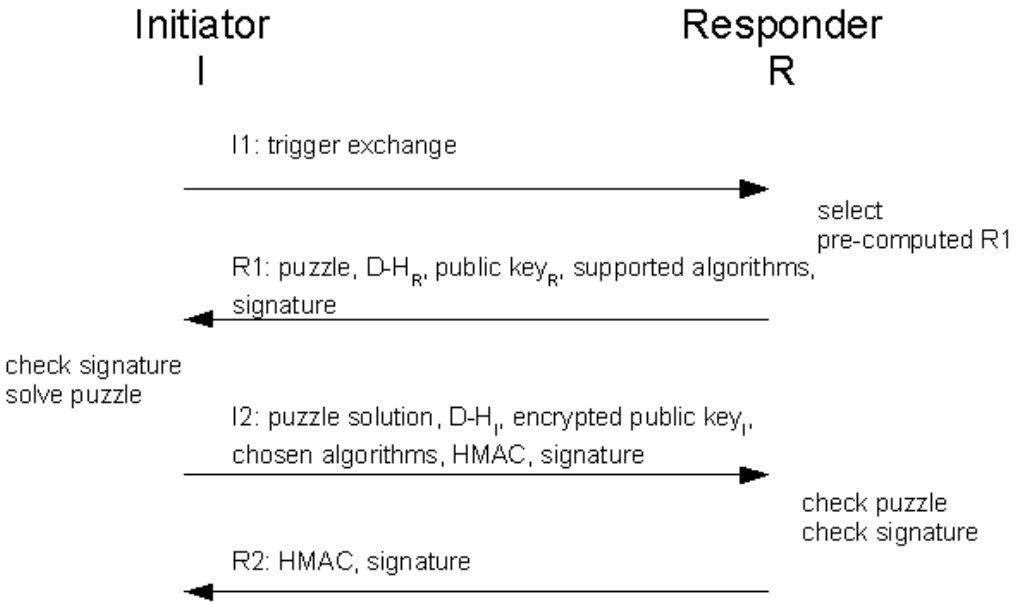


Figure 2-1. HIP Base Exchange depicting the exchanged messages and their contents

The third message, I2, is used to convey the solution to the puzzle back to the Responder, so that the Initiator can prove that it has invested some computational effort. Additionally, it consists of the Diffie-Hellman parameters of the Initiator, the used cryptographic algorithms, host identity of the Initiator, Hashed Message Authentication Code (HMAC) and a signature. The host identity can be encrypted using the algorithm specified in the message and the key computed from the Diffie-Hellman parameters, so that the identity of the Initiator remains hidden from the passive eavesdroppers and cannot be tracked, although [Aur05b] argues that the encryption is unnecessary and potentially harmful when it comes to stateful middlebox functionality along the path, because the middleboxes do not learn the key nor can they then perform operations such as signature checking. In addition, tracking can be done based on the HIT, which is available in the messages. After checking the puzzle solution the Responder proceeds with verifying HMAC, i.e., the result of a keyed hash function, and the signature. As a final step of the exchange the Responder sends R2, which contains just the HMAC and the signature, but it is needed in order to protect the Initiator from replay attacks, which could otherwise result due to the fact that the signature in R1 is precomputed. In other words, the binding between the public key and the exchanged Diffie-Hellman parameters is provided. After R2, the exchange is complete and the data packets can flow, potentially secured with the negotiated security association (not shown in the figure).

Thus, with this kind of procedure the parties have been able to establish an identity association, that is, they learn each other's public keys, and create keying material, which

can be used to protect any further communication between them. In essence, we have secure names for the participants of the communication, making it easier to refer to them in a secure fashion in any subsequent interaction. This also includes attaching statements to those names, that is, authorisations, but the identified entity could also issue its own statements about other entities.

3. NETWORK ATTACHMENT

3.1 Network attachment in brief

Network attachment is the process of connecting two nodes so that the communication can start flowing between them and other off-link nodes can be reached. In a sense, this can be called the bootstrapping of communication. A typical case is when an end device, a client, connects to an access point and requests connectivity services, but an evolved scenario might involve nodes, which are representatives of their respective networks, i.e., networks attach to each other. However, some other entities might be involved in this process, too. This could include additional devices in the edge network, such as an access concentrator, which controls several different access points and provides interworking services. Additionally, there could be separate authentication entities, which are responsible for the authentication and authorisation of the end device. They could be located quite far away from the actual point of access, provided by the visited network, as they could reside in the home network of a roaming client. Also, sometimes authentication might not be mutual, that is, only the client gets authenticated, and even if the home network is authenticated, it is another thing whether the client actually has explicit notion of the authenticity of the visited network.

Note that the previous discussion basically relates to asymmetric cases, i.e., the client and the access point are not equal in their communication capabilities. Another viewpoint is given by ad-hoc scenarios, where the relationship is more symmetric, but we largely leave that out of our discussion because for us the ultimate goal of attachment is interworking across different networks, that is, inter-networking. Note, however, that the envisaged future network scenarios, for example, ambient computing can exhibit more ad-hoc like properties, but there is still assumed to be some sort of interconnection point for accessing core networks.

The attachment procedure consists of ordered steps that need to be executed before inter-networked communication is possible. Typically, one has to first be aware of the other node, so that the link layer connectivity can be set up, i.e., link attachment can take place. The initiator of the communication can try to initiate a suitable query procedure to discover the possible counterparts or the other party, typically the access point, can broadcast information about its existence and the services it provides. After the discovery process, there might be some additional negotiation taking place. For instance, the participants could negotiate a security association in order to protect their link. In

wireless LANs, this could be done with the help of 802.11i, which is discussed later in Section 3.5.1. Once the link is set up, additional procedures are still needed to configure the inter-networking connectivity. Generally, this means acquisition of an IP address and finding a node which can route packets so that other nodes can be reached.

While this could be manually configured, it is not a very scalable way. Thus, nowadays automated approaches to configuration are favoured, so that no human intervention is required. In a stateful configuration approach, Dynamic Host Configuration Protocol (DHCP) with centralised servers is often used, especially in the IPv4 networks. In DHCP, the client receives an address from the server. The stateless configuration approach, on the other hand, relies on the ability of the client to form its own address, even though the network generally provides some support for this. This is commonly used with IPv6 networks, in which the routers broadcast their network prefixes, so that the clients can use that information to create an address. The downside for this is that the clients are also responsible for making sure that the created address is not already in use. Thus, duplicate address detection (DAD) needs to be run, which can cause some delay, up to a second or more [Tho06], before the actual communication can start. Of course, using stateless configuration does not mean that one would not be able to use, for instance, DHCPv6 [Dro03] as well to get some additional information. It is also worth noting that in IPv6, one should check the existence of duplicate addresses even when using DHCP [Dro03]. A similar duplicate checking procedure is suggested for IPv4 as well [Che08]. The details of these two different configuration approaches are discussed in more detail in Section 3.3.

There could still be some additional security measures that need to be taken care of. It might be that the client only has limited connectivity and needs to be authenticated and authorised on a network level to attain full connectivity. Protocol for carrying Authentication and Network Access (PANA) is one such measure [Jay08], even though it is not in widespread use. It could also be that IP security (IPsec), a security framework defined for IP, is required in order to meet integrity and confidentiality requirements, so steps are needed to set it up, for instance, by first running Internet Key Exchange (IKE) to establish the authenticity of the attaching parties and the needed session keys. As mentioned, these steps usually require contacting external entities, which are able to authenticate the client in question. Quite often protocols like RADIUS and various Extensible Authentication Protocol (EAP) methods are used for this. However, it is also possible to have an opportunistic approach, in which identifying keys and the corresponding proofs-of-possession are exchanged, but the real identity of the other party might not be known. This basically just ensures the sameness property, that is, the nodes can identify entities based on their keys and be sure that they are communicating with the same entity at different points in time. SSH is one example of this approach as mentioned in the previous chapter, even though it is not a network attachment protocol per se. More information about these security mechanisms are given in Section 3.5.

3.2 Threats and attacks

Various threats exist specifically within the attachment procedure. One obvious threat is the interruption of link layer communication, when a shared medium is used, which is naturally more insecure than, for instance, physically secure switched links. Thus, one can try to congest the medium in various ways. Other kinds of more subtle DoS threats are possible as well, especially when the attacker wishes to remain hidden. The accessing node might be fed with fabricated or modified configuration information or prevented from getting it, effectively making it unable to communicate with the outside world. Similarly, the nodes responsible for providing configuration could be flooded with information so that they would not be able to respond to legitimate requests. In the case of DHCP, one could simply deplete the server of the available addresses by making several requests with spoofed identities. Spoofing of an identity provides many ways of attacking the other nodes, some of which can result in DoS conditions. The DAD procedure can provide a venue for this, for instance, as detailed in Section 3.3.2.

A more ingenious attacker might try other approaches in order to control the communication without being detected or just steal the service. This could involve redirecting the packet flows. For instance, a node might pose as another node and get the traffic intended for that node to itself. Similarly, it could even pose as the gateway and get all the traffic. The traffic could be then forwarded towards the correct end point, so that it might seem that everything is working as expected. This is an example of a MitM attack.

A MitM attacker can try to modify the communication, but can be content in eavesdropping, if no protection measures are in effect. Even if protection measures are in place, but implemented poorly, for instance, with no authentication, the attacker can still subvert the communication. Another interesting point for MitM is behind the access point, i.e., in between the nodes, which are responsible of deciding and enforcing the access control. Thus, in case integrity and confidentiality protection mechanisms were not applied, one might try to change the access decisions or snoop, for instance, key information, which was intended to be used between the access point and the end point. If this were successful, even protected communication could be intercepted and listened to. This emphasises the need to consider the attachment procedure from a holistic perspective, so that weak links are not introduced in other parts of the system.

3.3 Configuration approaches

As the address configuration is the central piece of network attachment, we briefly go through the different approaches in automated configuration provisioning that are in use nowadays.

3.3.1 Stateful autoconfiguration

The essential feature in the communication of today is IP connectivity due to the dominance of IP as an inter-networking technology. So, one of the basic requirements for the network attachment procedure is the configuration of the IP layer, that is, provision of an IP address and other network specific parameters, such as network prefixes. In the past this involved manual configuration, but as the networks have grown in size and the number of nodes increased the management has become burdensome. Therefore, dynamic procedures, like those provided by DHCP, have become popular. It enables controlled provision and management of the configuration information, thus increasing the scalability of the networks. Additionally, it better supports the mobility of the nodes; at least from the nomadic perspective as that does not support seamless sessions across different points of attachment. Due to this managed approach it is sometimes called stateful autoconfiguration, i.e., state information about the individually configured nodes is stored in the servers.

DHCP has been defined both for IPv4 and IPv6 networks, but quite understandably the IPv4 version currently has the major part of the deployment. It is based on four different messages, which are exchanged between the client wishing to receive configuration and the server that is providing them [Dro97]. The exchange can also happen through a relay server, so it is not necessary to have a DHCP server on every link. Figure 3-1 depicts the basic message flow. In the first phase the client uses the DHCPDISCOVER message to find the servers that are providing configurations. The message is sent as a broadcast, because the client does not know the servers at this point. In the second phase the servers that have received the discover message reply with a DHCPOFFER message, which includes the available network address for the client. If there are several DHCP servers, then it is possible that all of them will send this reply, therefore it is up to the client to decide which configuration information to accept and then send a DHCPREQUEST message with the information received. This message is also broadcasted, so that the other servers can also notice which server was chosen. After this, the selected server can confirm the parameters by sending a DHCPACK message to the client. In certain cases when the client already has some idea about its address, for instance, when it is renewing its old address, it is possible that the message exchange only contains those last two messages without having to go through the discovery process.

The reason for having this many messages in the full procedure is the incentive to prevent different clients from receiving the same configuration and at the same time prevent the unnecessary reservation of server resources, i.e., available configuration information. The various messages can contain several different types of information and not just the assigned network address. This could include, for example, addresses of Domain Name system (DNS) servers and routers [Ale97].

The IPv6 version of DHCP is sufficiently different to make it hard for the different versions to interoperate as it defines some new messages and a completely different message format [Dro03]. The basic concept of the client and the server exchanging four messages to complete the configuration procedure remains the same, though. There has been some work to unify the representation of the identity of the client in DHCPv4 and DHCPv6 [Lem06], but it does not address any security issues per se.

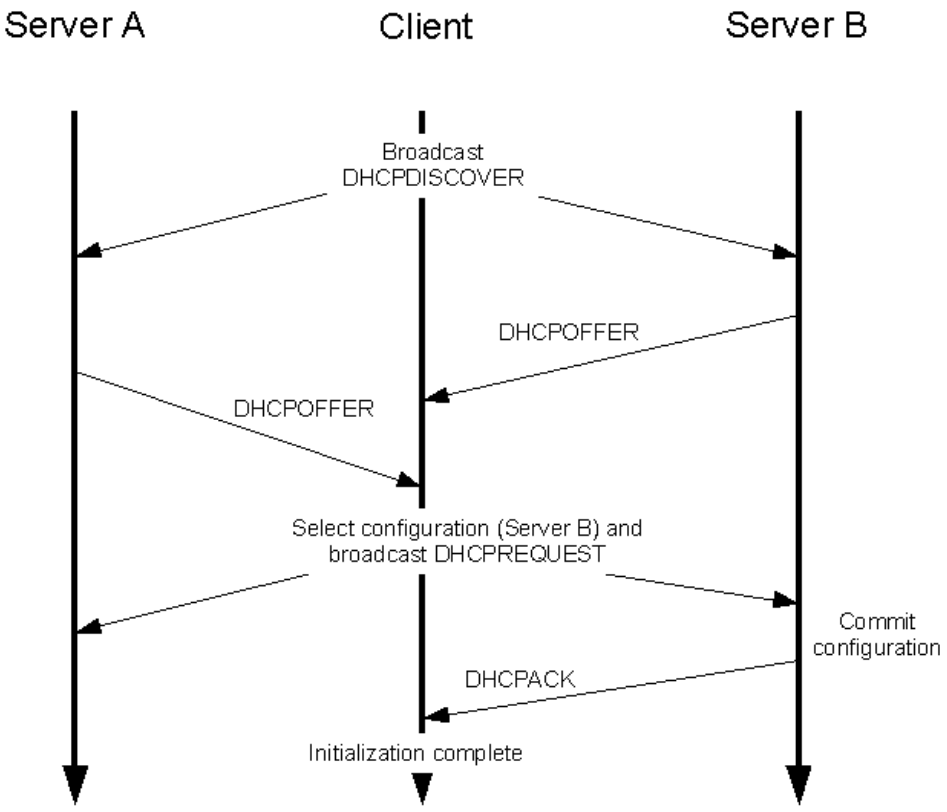


Figure 3-1. Basic DHCP procedure describing the message sequences between the client and configuration provisioning servers

The security in DHCP leaves much to be desired. Especially the IPv4 version does not provide any security at all, even though an additional specification for message authentication exists [Dro01]. In essence, in a typical DHCP environment it is quite possible to provide false configuration to the client or make the server waste its resources, which may result in DoS conditions. Instead of having a separate specification for security, DHCPv6 specification has incorporated the security mechanisms to itself, but it still provides only rudimentary security. There is no confidentiality protection, even though it can be argued, as the specification mentions, that configuration information does not need any confidentiality. There still might be some security sensitive material that could benefit from the protection, even the specification requires the use of a reconfiguration

key, which is used to authenticate subsequent reconfiguration messages from the server and which is sent in clear. DHCPv6 also does not authenticate the first client message, so it would be quite easy to deplete the server of its available addresses. This is further made worse by the possibility of using a rapid commit option, which enables the participants to complete the configuration just by using two messages: client request and server reply. In other words, a malicious client could make the server believe that the client has taken the address into use even though it has not. Perhaps the most major issue with DHCP authentication is that it is basically intended for an intranet environment, which is controlled and manually configured by a single administrative entity. From the point of view of scalability the manual configuration is never a good thing. This is likely the reason why the authentication scheme is not deployed in real world scenarios.

3.3.2 Stateless autoconfiguration

IPv6 makes it possible for the node to configure its own address [Tho07]. The procedure can take place quite independently in the absence of any routing devices as the node can form its own link local address with a known prefix. Of course, in this type of case one cannot expect to get connectivity beyond a local link. If at least one router is present, it can advertise subnet information to the client, in which case the client can form an address that can be used outside the local scope. It is worth noting that there also exists a procedure for the IPv4 host to configure its address in the absence of DHCP servers, but it is limited only to local link communication and uses a fixed subnet prefix of 169.254/16 [Che05].

The autoconfiguration process takes place in cases such as when the interface of a client attaches to a link or the system is initialised after start-up. The first phase is the formation of a link local address, which combines a well known link local prefix and an interface identifier. After that the client can either listen to the router advertisements on the link or try to solicit a response by sending a router solicitation message. The advertisements contain information about the used subnet prefixes and some additional information, such as instructions to use stateful configuration instead [Nar07a]. Combining the received subnet prefix with the interface identifier can result in a global address. Further stateless information, such as a list of available DNS servers, can be received, for example, with the help of stateless DHCP [Dro04].

Because several clients could be performing the address generation on the same link at the same time, it is possible that different clients end up having the same address. Therefore, it is necessary that the clients always run a DAD procedure before assigning an address to the interface. This happens with the help of a neighbour solicit message that contains the tentative address and is broadcasted on the link, or more specifically, it is multicasted to the group specified by the tentative address. In order to mitigate the possibility of several nodes which have booted up at the same time, sending solicitation

at the same instant, the node is required to delay the operation with a random interval. If someone is already using the address, it will reply with a neighbour advertisement. In case two clients are trying to configure the same address at the same time and both notice the solicit message, they have to refrain from using that address. According to the specification the client ought to resort to manual configuration at this point, unless the client has been preconfigured with other interface identifiers as well. However, there exists a specification for employing random identifiers instead of the fixed identifiers [Nar07b]. This also takes into consideration the privacy implications of using fixed identifiers. In other words, if the client is using the same identifier in different subnets, it is possible to track the client movements.

One could of course ask whether DAD is really needed and whether the consequences are severe, if it is skipped. Basically, with 64-bit interface identifiers, the probability of having two of the same addresses can be seen to be quite small. However, some network card vendors might not always follow prudent practises of assigning unique MAC addresses, which usually contribute to the formation of interface identifier. Thus, an address collision would be quite possible. In that case, the connectivity of the parties having the same address would be degraded. This can actually be worse than some other temporary network failure (like what might happen somewhat frequently with mobile networks), because there might not be a recovery option for it (until the conflicting nodes would change the networks).

It is worth noting that the duplicate address discovery may result in DoS. It is quite simple for some other node to respond to the neighbour solicit messages, thus preventing the legitimate client from forming an address. The earlier versions of the specification suggested the use of IPsec for securing the link local traffic, but in the environment where the nodes are mobile and under the administration of different authorities, it may be hard to establish trust between the different nodes, so that IPsec could be meaningfully deployed. In essence, the key management is problematic [Nik04a]. Nowadays, [Tho07] suggests using Secure Neighbor Discovery (SEND) protocol instead, which is discussed later in this chapter.

3.4 Requirements for configuration provision

When considering the configuration provisioning, which is the important part of network attachment, one needs to address the aforementioned threats in some way. Thus, one can set the following requirements, especially if one were to design new configuration methods:

- Integrity, authentication, and confidentiality of the messages should be provided
- Flexible authentication and authorisation models

- Privacy protection should be possible
- Denial of service should be addressed
- Efficiency
- Key management

Authenticity and integrity of the messages are imperative. Otherwise, as discussed above, a malicious party could modify the messages at will and direct them to an unintended party. The confidentiality of the configuration messages is not generally considered to be that important [Dro03], but in the future networks the proliferation of different kinds of services could change this. From the authenticity perspective, the sameness property is also an important requirement. In other words, there is assurance that the other party does not change during the exchange or across exchanges.

This also allows more flexible and usable authentication models, that is, a "better than nothing" kind of security [Tou08]. With suitable identifiers, like crypto-ids, authorisation needs can be better served as well. With such identifiers one can have more fine grained policies regarding the allowed actions. This way explicit authorisation can be demanded instead of expecting that authentication of identity implicitly authorises actions. Additionally, one should strive to decouple the authentication and authorisation as this can be used to provide better privacy protection, for instance, to disallow the tracking of a user. This is especially evident in cases where multiple parties are involved in the transaction, and assured statements about the properties of a party need to be made.

DoS is an increasing concern nowadays and as shown previously, can take place at many different levels. The motivation can just be intention of doing something malicious (say, because of a disagreement, or political motivations), but quite often it is a question of trying to get some financial benefits. In any case, the financial losses for the victim can be considerable [Dub04]. Thus, any protocol should take the DoS into account in the design phase. While it may be hard to provide defence against certain DoS attacks, like destroying the physical link, and sometimes it is hard to discern the features that might be potential DoS sources, the design process should already consider the potential mitigation measures at the early stages.

The utmost efficiency may not be that important with fixed lines broadband connections, but the situation is entirely different in the mobility scenarios. Especially if the protocol employs several roundtrips, a long latency in the wireless link can severely impair its operation. This is also problematic in the case of fast moving mobiles, which change their points of attachment often, i.e., make many handoffs. Therefore, it would be beneficial to keep the roundtrips to a minimum. An additional efficiency consideration should relate to the whole design process, which should include the security meas-

ures right from the start. Too often it has happened that the security is an add-on or an afterthought, causing security and usability problems [Yee04].

While it may not be so evident from the above discussion, the key management is also an important part of the overall solution. The security design might just state that IPsec is used to protect a solution. However, this really does not yet say how the actual keying material is transported in a scalable way. For instance, as indicated above DHCP has security options for authenticity, which basically rely on manual configuration, but it has really not been deployed because of the scalability issues introduced by the manual configuration involved [Hib06]. Also, it is not sufficient to say that PKI will solve this, unless one can be certain that such infrastructure is available for the solution to be designed. As we have seen, truly global PKI with equally approved trust roots does not exist.

3.5 Security mechanisms

This section goes through some of the security issues and mechanisms, which are relevant to our discussion with respect to the attachment procedures.

3.5.1 WLAN security

The focus of this work is at the network level, but as wireless LANs are already used quite extensively, we mention here some of the security features used for WLANs as some of the mechanisms, such as EAP, share commonalities with our other discussions. Namely, we refer to 802.11 networks, which are operating in infrastructure mode and include an access point and one or more clients wishing to gain connectivity. It is not uncommon to run WLAN in unprotected mode and the original security mechanism, Wired Equivalent Privacy (WEP) can still be used as a protection measure. It uses a shared secret based approach to provide the link with encryption, basically aiming to provide the same level of security as a fixed line would. However, WEP has long since been shown to be vulnerable [Bor01], therefore mechanisms such as those provided in standards 802.11i and 802.1X are recommended for providing link layer security in order to form a Robust Security Network (RSN). They are sometimes referred as Wireless Protected Access (WPA) or WPA2, even though that is a promotional term coined by the Wi-Fi Alliance, an organisation geared to promoting WLAN interoperability and certification of products.

802.11i is effectively an amendment to the original 802.11 standard, which has since been integrated into the 802.11 specification [802.11]. Along with the enhancement of link protection features, it also subsumes the functionalities of 802.1X to provide port based access control functionalities. This allows introduction of external entities into the authentication and key generation process with the help of different EAP methods. However, 802.11i also includes a mode of operation which is intended for small envi-

ronments, like homes and offices, and only requires a pre-shared secret between the client and the access point. Weak passwords can pose a danger, though [Mos03]. Some other weaknesses have been detected as well in the use of WPA [Tew09]. Additionally, management frames could still be in danger, like disconnecting unsuspecting clients. A recent standard, 802.11w, intends to extend RSN to protect management frames as well [802.11w].

When taking advantage of 802.11i and 802.1X, the WLAN attachment procedure basically works as follows; after the initial association, an EAP method is initiated so that the access point functions as a relay to the messages exchanged between the client and the authentication server. This demonstrates the benefit of EAP, which enables flexible use of different authentication methods irrespective of the capabilities of the access point (other than understanding the basic EAP) [Abo04].The EAP messages between the access point and the authentication server are then typically run on top of the RADIUS protocol [Abo03], although other Authentication, Authorisation and Accounting (AAA) protocols could be used as well. Even though the nodes are identified by their MAC addresses, the client also has to be able to provide an identity, which the authentication server is able to authenticate. For instance, exchange of certificates could take place when using EAP-TLS as the EAP method [Sim08]. The outcome of this process gives indication to the access point whether authentication was successful. Additionally, keying material can be created which can be used to generate a suitable session key between the end node and the access point. Naturally, the authentication server also needs to transmit this information first to the access point. Only after the procedure there is credibility to the SSID, which the access point uses to identify its network. Thus, this emphasises the need to have proper protection between the access point and the authentication server. After the parties are in possession of the session key, they also have to run a 4-way handshake procedure, with which they are able to ascertain that they are in the possession of said key.

The process is illustrated in Figure 3-2 (adapted from [802.11] and [802.1X]), which uses EAP-TLS as an example of the EAP method. The figure shows the different phases of the attachment: link association, entity authentication, and handshake for fresh key derivation based on the available keying material. In the last phase, the liveliness of the participants is ensured with exchanged nonces and Message Integrity Code (MIC) is used to ensure the integrity of the messages. In case no security was desired, i.e., the traditional open WLAN access, only the first six messages would be needed. The figure does not give all the details of the individual messages, but just tries to give an impression of the amount of messages exchanged and how EAP works (EAPOL refers to EAP Over LAN). Note that the depicted case is idealised, because, for instance, EAP-TLS might require fragmentation of too large messages, thus increasing the amount of messages exchanged.

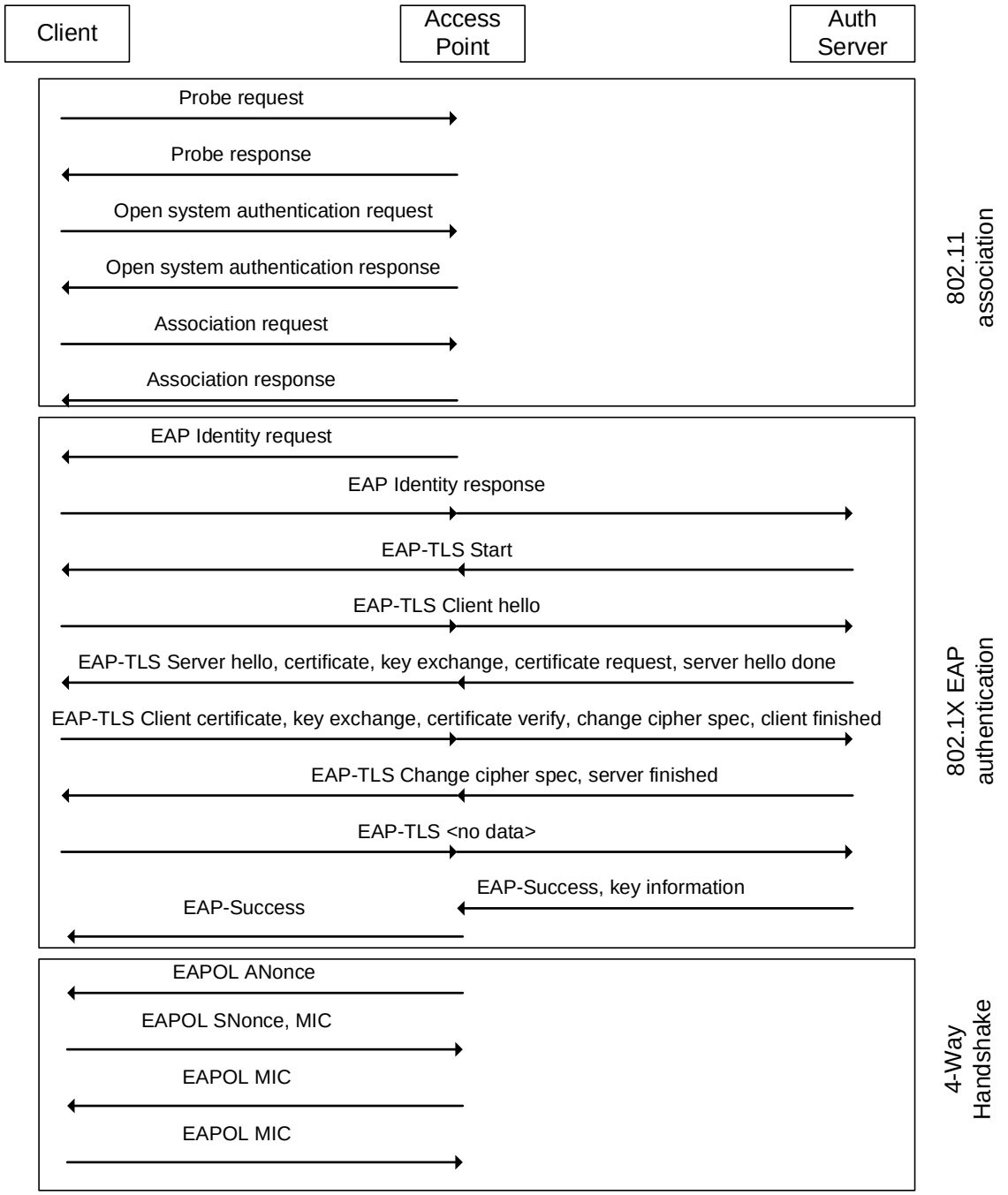


Figure 3-2. 802.11i protocol example showing the initial link layer attachment, authentication, and link layer protection establishment phases

The above example assumes that the client is in possession of a certificate, which can be presented in the course of EAP-TLS exchange. However, the parties are still identified by their MAC addresses, even though the calculation of a common link key takes these addresses into account, thus making it harder to spoof identities. Thus, the access control is enforced through the knowledge of the key. Generally though, the authentication process takes some higher level identifier without any reference to the link level and, in fact, the access point is not given any authenticated identifier from the point of view of the client. IEEE has also produced a standard, 802.1AR, which defines the possibility of

identifying devices through cryptographical means, although this basically means authentication is completed with the X.509-based certificate stored in the device [802.1AR]. The naming of the device is not that clearly defined, as long as it is a unique Distinguished Name (DN), for example, a device serial number. Thus, referring to the device outside mutual exchange, for instance, to assign authorisation or delegate rights, would not be that straightforward, that is, additional validation steps would be needed to ensure valid bindings. Especially considering that the certificate extension `subjectKeyIdentifier` [Coo08], which could have defined the used key in a concise manner, e.g., a hashed representation, is not used.

3.5.2 Protocol for carrying authentication for network access

The current IP level attachment procedure used in the Internet does not provide authentication services. Protocol for carrying Authentication for Network Access (PANA) is a network-layer proposal for providing access control between a client and the access network [For08]. It basically offers a transport for running EAP over IP, even though UDP is used. Thus, it aims to be link-layer agnostic [Jay08]. As it uses EAP, it bears logical similarities to the 802.11i case presented earlier. The basic operation dictates how the PANA Client (PaC) authenticates itself with PANA Authentication Agent (PAA) with the help of a suitable EAP method. The method can possibly generate keying material and ensure freshness by expecting the parties to exchange nonces. PAA can additionally use a separate authentication server, for instance, with RADIUS. After authentication, PaC is able to access other networks through Enforcement Point (EP), which is responsible for ensuring the proper filtering of traffic.

PANA basically expects that the client already has an IP address, but the client is expected to get another IP address after the authentication. Thus, PANA just concentrates on the authentication part and not so much on preventing spoofing of identity at a later state. However, there exists a PANA draft [Par05], which defines mechanisms to employ IPsec as an access control solution between PaC and EP. It basically expects them to run IKE or IKEv2 between themselves and use the keying material created during the authentication exchange to ensure authenticity. Thus, combining both methods, one could draw an analogy between 802.1X and PANA and between the 4-Way handshake and PANA IPsec.

Even though basic PANA assumes that the client is in possession of an IP address, there is also a draft for IPv4 cases, where the address needs to be configured first, even though it assumes that PaC and PAA are on the same subnet [Yeg11]. The client is expected to use unspecified address 0.0.0.0 and broadcast its messages (or use link layer unicast, if possible). After the normal PANA procedure, DHCP is used to assign the proper address. However, the draft does not discuss whether the DHCP would be somehow bound to the authentication procedure (as it should). So, in essence, spoofing and theft of service might be possible. Thus, address ownership is not addressed.

3.5.3 Neighbour discovery

When a node wishes to attach to other nodes, it first has to know which nodes are reachable through the link layer, i.e., local hosts, and which nodes are able to route information to other nodes, i.e., routers. This also includes address discovery, that is, in order to be able to send IPv6 traffic to its neighbours the node needs to learn their link layer addresses. These procedures are generally called neighbour discovery and IETF has defined Neighbor Discovery (ND) protocol to accomplish this in IPv6 networks [Nar07a]. This can also include procedures that relate to the autoconfiguration of IPv6 nodes and, like ND, it works through different message types of Internet Control Message Protocol for IPv6 (ICMPv6) [Con06]. In IPv4, some of the ND functionality was available through the use of Address Resolution Protocol (ARP) [Plu82] and ICMPv4 specifications [Pos81][Dee91].

As the security mechanisms for those definitions are largely left unspecified it is quite easy for a malicious entity to modify packets, make the traffic flow to unexpected places or cause DoS in several different ways. Many of these threats were briefly mentioned previously and are detailed in [Nik04a]. IPsec is offered as one alternative to authenticate the traffic, but the configuration of security associations ought to be manual, since the key management protocols like IKE cannot be used without first having some connectivity, resulting in a so called bootstrapping problem. There have been research efforts to provide security for ARP, such as Secure ARP (S-ARP) [Bru03] and Ticket ARP (TARP) [Loo07], which rely on public key cryptography to create a binding between MAC and IP addresses, but require a certain amount of preconfiguration. Similarly, preconfiguration is needed in the Cryptographic Link Layer (CLL), which already protects Ethernet frames and makes an effort to integrate it with DHCP [Jer08].

SEND tries to address some of the shortcomings of the IPv6 discovery process [Ark05]. The mechanisms SEND uses to protect the traffic are certificates, signatures and Cryptographically Generated Addresses (CGA). Additionally, the messages include a nonce and timestamp to mitigate the threat of replay attack and to ensure that the messages are fresh. Certificates are used to authorise routers to advertise certain address prefixes. In other words, some trusted entity, like an Internet Service Provider (ISP), has certified the router to be in charge of routing the specific address range. The integrity and authenticity of the different messages are ensured with RSA public key based signatures, which can be either related to the public keys exchanged in certificates or used with CGAs. CGA is a cryptographic mechanism for ensuring the ownership of a certain IP address and is discussed in the following section.

A problem that different nodes employing SEND may face is that they cannot determine a common trust anchor for IP address authority. This way the authorisation statements have no value to the other party. Also, SEND does not specify the case, where the nodes generate their addresses using fixed interface identifiers coming, for example, from their

link layer identifiers. It is, however, suggested that some sort of authorisation certificates could be used there as well, but they are not specified. DoS attacks are still possible too. For instance, an external node could try to make the router query for non-existent addresses in its local link. This could severely impact the performance of the local link [An07]. Protocol expects the implementations to take appropriate precautions, such as rate limiting and caching of existing addresses.

3.5.4 Address ownership

The spoofing of addresses in the current Internet architecture is relatively easy, because there are no mechanisms in use that would bind the address to the actual identity of the sender. This can result in various kinds of impersonation and DoS attacks [Nik01]. In order to be able to prevent such activity, one needs means where it is possible to prove they are in possession of a certain identifier, i.e., an address. Such proof can be provided by cryptographic means, so that with very high probability, only the legitimate sender could have calculated and formed the correct proof. One such proposal is the Cryptographically Generated Address (CGA), which uses public key information in the formation of an IP address without having a TTP [Aur05a]. Similar ideas were also presented in [OSh01] and [Mon02]. It is also possible just to send some opaque data to the target and if the same data is received, it is considered as a proof that the target is able to receive traffic to that specific address. Even though this kind of return routability check does not rule out the possibility of exploitation, it does decrease the amount of potential attackers, because they would have to be able to actively monitor the traffic and be prepared to respond to such queries. Thus, in essence, they need to be on the path.

CGA is only applicable to IPv6 and, more specifically, to addresses that are formed from a 64-bit subnet prefix and 64-bit interface identifier parts. This format is also used by most of the global unicast addresses [Hin06]. Because the subnet prefix is normally received through router advertisements, CGA generation only concerns the interface identifier part. The generation process first basically uses a random number and the public key of the node to create a hash value, which has a suitable amount of zero bits. The next phase of the process takes the subnet prefix, the public key, and the aforementioned hash value and calculates another hash value, which is used to create the interface part. The computational difficulty of the calculation is dictated by the amount of required zero bits in the first hash, so much of the “hard” calculation can be done in the first phase without knowing the subnet prefix. After the address is ready, the process still has to check the uniqueness of the address by performing duplicate address detection, which is, making sure that no one else is using the same address on the same link.

CGA is not used on its own, but with some other protocol, like SEND (see previous section). The other protocol is responsible for defining a mechanism to include a CGA signature with it, so that the receiver of the message is able to compute that the key used to sign the message is also the key that corresponds to the IP address of the message.

The receiver knows the public key because it is transmitted in the signed message along with other parameters used in the CGA generation.

Even though the generation process of CGA makes it possible for an adversary to calculate exactly the same address, he still cannot impersonate the legitimate user, because he also ought to be able to sign messages with the corresponding private key. However, if the adversary is able to break the second preimage resistance property of the hash used to calculate the interface identifier, that is, find another input that hashes to the same output, he is able do the impersonation with the public key in his possession. Obviously, such weak hash functions should not be used. An additional security feature is the CGA generation process, which ensures that the verification of the CGA is relatively simple with a couple of hash calculations, whereas creation can take time. This way the potential attacker may have to invest too many resources, thus making the potential impersonation infeasible, for example, in the case of DAD DoS attack.

From the privacy protection point of view, CGAs do not provide very much protection, even though the node can generate new addresses with the same public key, but the signature used, for example, in SEND, will contain the public key, thus making it easy to track the user, unless the keys are changed from time to time. The keys, however, are generally not linked to any real identity, because CGA does not require any infrastructure, such as PKI, to be able to work. This is due to the fact that in this scope it is more important to know that the specific address is used by a specific entity than to know who that entity is. Also, as the lack of global PKI has shown, relying on such structures can severely hinder the deployment of the solution. However, as proposals such as DNSSEC [Are05] are taken into more widespread use, synergy benefits can be achieved. Other approaches with CGA have also been proposed, like integration with IKE [Lag07][Cas04]. A similar approach could be envisaged with HIP, as well. Additionally, multihoming cases are considered in [Bag09], even though that document also suggests a computationally less demanding alternative, Hash Based Addresses (HBA), which bind together a set of addresses without using public keys.

There are also some enhanced proposals, which enable the CGAs to be proxied, that is, some other node, like an access router, can prove the legitimacy of the address ownership on behalf of the actual owner. This is especially beneficial in mobility scenarios. One such proposal introduces Multi-key CGAs (MCGA), which are created using a group of public keys [Kem06]. A corresponding signature, or so called ring signature [Riv01], can be created by any node in the group, but the verification requires all the public keys of the group, thus protecting the privacy of the node. Naturally, one could envisage a case where the proxy device would be responsible for creating the address (after receiving the proper interface identifier of the client) and communicating that to the client with, for example, DHCP. Another thing is whether the private key needs to be communicated as well, as it increases the chance of key compromise. There is some work which already considers CGA with DHCP, but the client is still responsible for

creating the address and the DHCP server is responsible for acknowledging whether the address is suitable for the network in question [Jia11]. The possibility to combine CGA with HIP and DHCP was also briefly mentioned in [Hei04].

Other slightly similar approaches, although emphasising the accountability aspects, include Packet Level Authentication (PLA), which suggests including an elliptic curve based signature and the corresponding public key to every packet [Can05]. The integrity of each packet can then be verified and the legitimacy of the sending node can be ensured through the accompanying TTP certificate, even though in lightweight mode it could be omitted [Lag10]. So, basically the ownership of the address is not ensured, but the integrity of the packet and the authenticity of the sending node are.

An even more drastic approach is taken by Accountable IP (AIP), which proposes architectural changes to introduce the accountability property to the Internet architecture [And08]. This takes place with the replacement of IP addresses with a flat address structure, which identifies the administrative domain and the end node in a self-certifying manner, that is, identifiers are basically hashes of public keys and in the possession of the administrative domain and the end node, respectively. The self-certifying nature of the identifier is used in the attachment procedure, which requires a separate verification process with the access point during which the node proves that it is in possession of the private key corresponding to the used public key. Naturally, such long term vision has deployment challenges. The same is true for other novel information centric architecture proposals, which rely on flat, self-certifying identifiers (for example, RTFM architecture [Sär08]).

TrueIP is a proposal, which does not include such a disruptive approach and instead suggests employing Identity Based Cryptography (IBC) to provide the proof of IP address possession [Sch09]. The IP address is used as the private key, for which the secret parameters are provided by the infrastructure and the corresponding public values need to be distributed to the recipients, as in typical IBC systems [Bon01]. Thus, the client is not truly in possession of its address, but has been delegated the right to use it. The approach is stateful and greater reliance on the infrastructure is needed, which also has to take care of the distribution of the secret value to the configuration provisioning points.

3.5.5 Host identity based approaches for network attachment

There are various proposals, which build their security on the existence of a host identity or other similar identity based on cryptographical properties, as can be seen from the previous discussion. The proponents of HIP especially expect that the future communication setup involves a handshake step for authenticating one's host identity. Thus, every entity would be expected to be in possession of such an identity.

HIP was proposed in the context of configuration provisioning already in [Hei04] and that work was further detailed in [P1]. This and the approach suggested in [Ark04b]

have evolved into various, more general network attachment protocol proposals, which take a holistic view to the whole attachment procedure. For instance, [Ark06] proposes how a HIP like attachment can provide a more efficient way to connect to the network and delegate signalling responsibility to the access devices. Similarly, [Kor07] suggests an enhanced attachment procedure for WLANs, based on the use of HIP. It also makes the observation that the performance of a solution, which provides the security on the network layer with HIP, is comparable to the case where the security is provided by the link layer functionalities, that is, the aforementioned 802.11i mechanisms. [Ark06] also demonstrates that quite many messages are needed for establishing IP connectivity (27 in the given mobility example), hence the identity based approaches can be used to increase the efficiency in terms of reducing roundtrips. As one can see from the previous Figure 3-2, the amount of roundtrips in WLANs is also considerable. [Rin06b] has a similar identity based approach, as well, and it suggests how a general Ambient Network Attachment Protocol (ANAP) could be implemented in various different layers (basically on layers from 2 to 4). Even though it does not go very deep into the actual protocol details, so the relation to HIP is not so obvious. However, it is worth noting that all of the above suggestions come within the ideas developed in the Ambient Networks EU project, hence it is understandable that there are similarities in the approaches. This is also close to the suggestions concerning the attachment procedure in information centric networks given in [Kjä09].

In the previous proposals one has to naturally ask how deployable the suggested enhancements are. Completely revamping something well-established, like current WLAN, is not a feasible approach. For new link technologies, such efficient designs might be more easily attainable, as stated in [Ark06]. If, however, technologies like HIP take off, enhancements could be incrementally introduced as the building blocks would be readily available. Thus, the WLAN association phase could be kept as it is and HIP could be used as the main security component, as demonstrated by [Kor07]. Additionally, other functionalities could be introduced by piggybacking them on top of HIP, for instance, EAP mechanisms could be used to provide support for a wide array of authentication mechanisms.

If we take a closer look at the configuration procedure and the mechanism proposed in [P1], we need to consider the discovery of proper configuration provisioning servers. In DHCP the case, where the server is on the same subnet, is somewhat straightforward because the broadcast (or multicast with IPv6) can be used to discover the parties. In case the server is not on the link, relaying agents need to be used and they may know the existence of a DHCP server through preconfiguration (like in centrally managed environments) or they need to broadcast the message onwards. If we are using the HIP-based approach for the configuration provisioning, we have several alternatives for reaching the target server. Firstly, the client might know a suitable HIT for the server. This might be received during the link layer attachment step, for instance, in a beacon message of an 802.11 network. The HIT could also be a sort of service identifier, which

would function in anycast fashion and describe a type of service, such as configuration provisioning. This would be close to the information centric networking ideas presented, for instance, in projects like Publish-Subscribe Internet Routing Paradigm (PSIRP) and that could mean employment of Bloom filters [Blo70] to perform routing based on the identifier, that, the filter would indicate the link where the message is to be sent. Basically the filter could be learnt from the access point or the filters would be located in the routing nodes and the configuration resource would be indicated by a well-known identifier. If we were to still retain our IP-based view point, also we would need overlay routing, that is, identity routing, for these kinds of identifiers. Host Identity Indirection Infrastructure (Hi3) [Nik04b] could provide one such possibility, as described in [P1]. In other words, we would have HIP capable hosts, but the signalling messages would be traversing an identity based indirection layer. It is worth considering, though, whether the networking environment is dynamic enough to warrant such complex discovery procedures, for example, the size of access network may not be very large. After all, in the typical managed setup the access point is aware of the configuration provisioning server.

However, one should also notice that even though the described procedure could provide extended security for the configuration provisioning, the client, if DHCP configuration steps were strictly followed, would still need to perform an ARP request (or duplicate address detection in IPv6) to check whether the address is usable. This might provide a venue for DoS. In IPv4 one natural approach would be the integration with previously mentioned TARP, but instead make a binding between the MAC and HIT. Naturally, an adversary could lie about its MAC already when communicating with the provisioning server providing the binding attestations, but this would mean that the adversary were not able to freely spoof any MAC. It basically ought to know a specific MAC which to target.

Previously mentioned CGA can be also used with the suggested method. In essence, the client could "outsource" the creation of the address to the server, which might have more computing power at its disposal. This procedure can take place when the server is in possession of the public key of the client, i.e., after the reception of an I2 message, and it is already aware of the used network prefix. The next message, R2, would provide the generated address as suggested in [P1]. While this approach benefits the computationally limited devices, one should still remember that in this setting the client is still expected to be able to perform public key operations. However, the real benefit would be the possibility to leave the duplicate address detection to the server, hence mitigating the effect of delays before the address can be taken into use. The server can keep track of the used addresses to avoid address collisions. In case there are multiple address provisioning servers in the area, they can communicate directly without resorting to multicast (assuming they belong to the same administrative domain and are aware of each other).

3.6 Authorisation aspects

An important consideration about the various service usage scenarios is whether the entity in question has a legitimate right to use the service or whether the service provider is authorised to provide resources for consumption in a certain context. In other words, even though the client might be authenticated, is it also authorised to request configuration information and get connectivity service? Another way of asking it would be whether there is a promise of payment. Similarly one can ask whether the access router, for instance, is allowed to provide the network configuration it is advertising. This issue has been considered, for example, in the context of neighbour discovery [Ark05]. While we talk here about authorisation, the term capability has also been used, especially in the realm of software engineering for expressing the access rights to an object [Mul86][Bel76].

There are many techniques for presenting authorisations and quite often they involve tokens of different kinds as they provide more scalable solutions. In smaller environments, like homes, one could rely on access control lists (ACL), which define the rights for the authenticated identity. For instance, in our context, one could base the authorisation on a certain HIT, so that a preconfigured HIT is allowed to make the connection or access a certain service. In a similar manner one could use passwords in such environments, but they possess a weaker proof of possession characteristic than the HIT based scheme, as passwords can always be guessed or stolen, for instance, using social engineering [Hei06]. One could also consider other stronger authentication services, like Kerberos tickets [Neu05]. It should be mentioned, though, that such Kerberos tickets could also include authorisation information, e.g., as is done in certain operating systems [Bre02]. This is based on the existence of a mutual shared secret, that is, secure envelopes are created and they can be exchanged between the known parties, albeit with limited accountability and easy delegation is not possible. Certain rights can also be based on past behaviour. For instance, in mobile IP one can use credit based authorisation, where the traffic with the new IP address is constrained by the amount of traffic exchanged with the old address [Ark07]. This basically tries to limit the amount of flooding that can be directed towards the yet unconfirmed new address.

As discussed in [P2], token based approaches are more interesting for the identity based solutions as they can be bound to the used identities. One such technique is Security Assertion Markup Language (SAML), which can make XML based assertions about the characteristics of an entity [OAS05]. As they are XML based, they tend to be lengthy, therefore not very suitable for size constrained network level solutions, which prefer unfragmented packet processing. However, one can give short references, called artifacts, to indicate the location of the actual assertion. This basically requires contacting an additional server to download it, hence extending the two-party communication to involve three parties.

Certificates embody similar characteristics and come in various flavours. X.509 certificates [Coo08] are perhaps the best known and are extensively used, for instance, alongside TLS to protect web sites. However, they are not necessarily the best choice for making authorisation statements because they concentrate more on entity authentication, that is, making a relationship between a name (a real world identity) and a key. There exists a separate profile though, which is more suited for binding different authorisations (attributes) to the identity. In this attribute certificate profile, the identity can be identified with a name, but it is also possible to provide it in the form of a hash of the public key of the holder of the attribute [Far10]. As the attribute certificate does not contain the public key, the two different certificates are expected to support each other, especially when noting that the lifetimes of authentications and authorisations can be different [Far10]. There is another similar profile for proxy certificates, which are intended for delegation, but they also overlap in functionality with attribute certificates [Tue04].

Simple Public Key Infrastructure (SPKI) certificates also resemble attribute certificates in their usage, but provide a less strict structure [Ell99]. In fact, as [Ell99] demonstrates, all the specifications have never been finished within IETF. Therefore, they can be used quite flexibly, albeit with questionable interoperability, and they have gained more popularity within the research domain, such as grids [Lag05]. Also, there are already well-established business models around X.509 certificates, i.e., there is less incentive for the stakeholders to promote alternatives. The security implication of this has been underlined by the events lately, which have portrayed less than diligent procedures for entity certification enrolment [Hal11].

[Ero04] observes relevance of payments to authorisations. Thus, one can also consider different kinds of electronic money approaches for making authorisations. After all, in essence, the resource provider is mainly interested in getting reasonable compensation for the resource usage. This can require a rather extensive infrastructure, which takes care of issuing electronic money tokens and arbitrating between the parties wishing to engage in a commercial transaction. The ease of copying electronic information sets its own difficulties for such a system, so that double spending and undue compensation claims can be prevented. On the other hand, one can also see this as a risk management problem such as depicted in [Bla02]. Much like in credit card business there is a risk involved, but with careful design and management it can be mitigated to an acceptable level. We come to these points in the next chapter about non-repudiation, even though we are not going to delve deep into the whole ecosystem of digital cash, we just briefly mention the possibility of using microchecks as authorisation tokens, that is, the ones employing KeyNote credentials [Bla02].

3.7 Analysis of identity based approach

If we reflect on the properties of the suggested host identity based approach in this chapter, it should be evident that much of the security is based on the security of HIP and the provided configuration information enjoys the same level of protection as the normal HIP packet [Mos08]. As such it provides DoS protection through the puzzle scheme for the server (Responder), which in our case can be seen as more important than the protection of the client (Initiator), because the server providing configurations is a more attractive target for an attack as it is more likely that the service is provided by a company or other business entity, which has economic interests at stake. Of course, it is still possible to try to use flooding to render it inoperational. The use of indirection infrastructure helps to mitigate this threat, but, as suggested by the HIP specification, the Responder should also protect itself by dropping similar I1 messages. One may question though how to detect similar messages, if no state is stored. The client should also avoid R1 storms, i.e., a large amount of simultaneous R1 messages, by checking that it actually has sent the corresponding I1, otherwise it could end up spending its time solving puzzles.

The client may face the threat of bogus servers, unless the server provides a statement, which authorises it to provide the configuration. In a wireless environment this could result, for example, in a case where the malicious server connects legitimately to the infrastructure, but makes the client pay for the access provided some dynamic compensation scheme is used, that is, the malicious server pays for its own access, but also charges the client for the access it provides. Another thing is whether this can be seen as a legitimate ad-hoc setting, where a more powerful node provides connections through itself to computationally limited nodes perhaps using some short range radio link, like Bluetooth. Generally, as in any HIP exchange, if no authentication or authorisation tokens by a TTP are used or the HITs are not otherwise known, the communication is subject to the threat of MitM. Lack of authorisation on the other hand, enables authentic parties to perform various attacks, like traffic rerouting, if mobility procedures, like mobile IP, are in use. One should note, though, the benefit of having better than nothing security to defeat the most casual passive attacks.

A malicious server could try to replay old R1 messages which contain some relatively static configuration information, but the client is able to detect this based on the puzzle generation counter coming from the legitimate server. The attacker could try to destroy the legitimate R1 messages in order to prevent the client from learning the current counter value, but this would not get the attacker any further than a typical DoS attack. The malicious client, on the other hand, may try to deplete the DHCP server of its resources, like available IP addresses, by continuously requesting new addresses. In order to do this, it would have to go through the HIP handshake procedure, which in itself is time consuming. Additionally, it would need to generate a large number of host identities or else the server would be able to detect the excessive requests. In case the host

identity remains the same, the excessive I2 messages can be filtered, in a case where the attacker first solves the puzzle and then sends messages containing invalid signatures with the aim of making the server waste its computational resources. The server should notice that the same identity has already provided a solution to the same puzzle. The server is also able to detect if the client tries to use replays of old I2 messages to request addresses, because the puzzles are likely to be too old. Naturally, when the server notices that it is under attack, it can increase the puzzle difficulty.

Privacy of the client is protected through the mechanism offered by HIP, that is, the possibility to encrypt the public key of Initiator. Although, as [Aur05b] states, this identity protection is ineffective if it is possible to run the exchange in the other direction as well and make the client assume the role of Responder, thus revealing its identity. Also, the identity is revealed to the server, so multiple servers could collude and track the client. The public key can be ephemeral, so that it is possible to discard it after a short period of time. This also changes the HIT, which otherwise could be used to track the client. Even if the key is temporary in nature, authorisation statements can be easily bound to it, thus promoting the decoupling of authentication and authorisation, that is, it is more important what actions are allowed to you instead of who you are. Such decoupling is also valid for permanent identifiers, even though the privacy concerns are more protruding depending how readily the information connecting authorisation, the key, and the name is available.

Here, one could also note the benefits of delegation, i.e., one could delegate authorisations and signalling tasks (or even pay on behalf of others) to other entities, when it is easy to bind such statements to an identifier used in another context. In case privacy is desired, the different identifiers need to be unlinkable to an outside observer. Thus, one needs to be able to convey the delegation statements in such a fashion that the identity of the delegator is not leaked, that is, confidentiality needs to be provided.

The parties would have to wait for the exchange to be complete and then communicate using ESP protected traffic, HIP UPDATE packets or use some additional payloads after the HIP header, in case the size of HIP parameters got too large. An alternative, which did not expect the parties to run full base exchange, would be the use of the HIP DATA mechanism described in [Cam11]. This could basically serve the same functionality as the authentication option found in DHCP [Dro01]. This does not make it possible to provide confidentiality to the configuration messages, but it can be used to provide integrity protection by including the hash of payload to the signature calculation. Note that even though it would be nice to use ESP protected traffic at this point, I2 cannot yet carry such payloads due to the fact that the Initiator does not know which SPI the Responder has chosen for itself [Jok08]. Also, lengthy packets may result in packet fragmentation, so the reassembly at the end points needs to be adequately resistant against DoS attacks that may result from this.

The nodes acting as relay servers do not function other than as forwarding servers. Therefore, they can only prevent the traffic or forward it to the wrong hosts. They are not able to modify the message, because they are protected by signatures. Also, they cannot claim to own the HIT of the other party, because they are unable to prove the possession of the relevant public key pair, unless they can break the key. Similarly, the relay server cannot inject additional packets, which could contain false configuration. If the parties do not have authenticity information about the HITs, the relay server could act as MitM. This can be prevented, if the parties are in possession of the authorisation information, as discussed earlier.

To summarise, the proposal is able to natively provide confidentiality, integrity and key management services. Even though the confidentiality of typical DHCP configurations may not have much value currently, the evolving network scenarios might prove differently in the future. The proposal also helps to mitigate the DoS attacks against the server. To some extent, the potential use of indirection infrastructure can provide additional protection against a flooding attack both for the client and the server, even though the full benefits come from the wide scale deployment of such infrastructure that may not be that realistic. Active MitM attacks are a concern, but they can be prevented with the use of security tokens provided by a TTP. This includes authorisation information, which is suited for ephemeral identifiers in privacy sensitive scenarios. Explicit authorisation also brings more granularities to the rights management, thus granting only the required privileges, that is, the so called least privilege principle [May91]. The use of authorisation tokens may require additional message exchanges though, and therefore have performance impact, unless more drastic changes to the basic HIP specifications are devised. However, in the basic form, efficiency is increased as the amount of round-trip exchanges can be reduced due the piggybacking of information on top of HIP handshake messages.

In this section we have seen many of the possibilities of host identities. Especially when we reflect on the requirements given in Section 3.4, we can see the benefits host identities provide for enhancing the security of the configuration provisioning, as summarised in the previous paragraph. More importantly, they enable secure naming, which allows binding of actions and attributes to the correct entities, hence mitigating the threats resulting from identity spoofing.

4. NON-REPUDIATION

4.1 Introduction

Certain transactions have the requirement that parties should not be able to deny, that is, non-repudiate, their involvement in the transaction. This could relate, for instance, to electronic commerce, where there is clear incentive to make sure that one party has paid and the other party has delivered the goods. A contract signing is also a traditional example: The participants prove with their signatures that they have seen the contract and approved it.

From the communication perspective, the transactions can take place either directly between the parties, the sender and the recipient, or with the help of a third party, delivery authority or agent [Zho96b]. The delivery authority, if such exists, is responsible for accepting incoming messages for forwarding to the correct recipient. An HTTP transaction between a browser and the web server can be said to be direct, but it is also possible to introduce a proxy element to act on behalf of the client. Another common example of delivery authority would be the use of mail transfer agents [Kle08]. Thus, there is a certain amount of trust to these third parties to deliver the messages.

The delivery authority can be viewed as an inline trusted third party, although a trusted third party (TTP) can also participate in the protocol without being involved in every message exchange. Based on the activity level of the TTP, it can be seen working either online or offline [Lou00]. That is, the TTP either actively participates in the protocol or is only required at the time of dispute. Offline TTPs could also be included in tasks like preregistration or other preconfiguration steps. [Kre02] additionally defines neutral and transparent characteristics for TTPs. A neutral TTP does not need to possess the knowledge of the information exchanged and a transparent TTP, if it is needed, produces evidence, which is indistinguishable from the evidence created by the participants. However, the TTP should be careful of what sort of operations it applies to data, which it does not know anything about. For instance, [Yan03] uses the public key of the TTP to encrypt the session key and it is expected that the TTP decrypts it before transmitting the key to the other party. If the process is automated and the TTP just blindly uses its private key to decrypt the given data, this could be clearly used against the TTPs and make it reveal confidential data. [Gur05] also shows how TTP can be tricked into decrypting data by reusing the encrypted data in a different transaction, i.e., the used secret key is not properly bound to the context.

As already indicated, TTPs can have several roles. Such roles include Certificate Authority (CA), notary, delivery authority, and adjudicator [Zho96b]. CA can provide authenticity and validity to the used keys and a notary can be used to provide assurance to the used evidence, which can also include timestamp service. An adjudicator is generally used in the case of disputes and it acts as a judge, which determines, based on the presented evidence, whether the non-repudiation policy was violated by either party.

When there is a need to introduce non-repudiation to the message transfer, the following characteristic can be required [Zho96a]:

- Non-repudiation of origin (NRO)
- Non-repudiation of receipt (NRR)
- Non-repudiation of submission (NRS)
- Non-repudiation of delivery (NRD)

NRO is intended to provide the proof that the sender has sent the message, thus it cannot later deny sending it. Similarly, NRR prevents the receiver from denying the reception of the message. NRS and NRD are part of the actions of the delivery agent and prove that it has accepted the message for transmission and delivered the message, respectively. In a sense one could think that within an indirect communication system, where the delivery system is considered to be trustworthy, just NRS and NRD are sufficient. However, the communication links can be unreliable, thus there is no real guarantee that the messages reach the intended recipient. Thus, in a typical direct communication model, NRO and NRR are more reasonable requirements to achieve accountability.

4.2 Fairness

Various ways of implementing a non-repudiation system can be devised, but it is another question whether the system is fair to all parties. In other words, it should not discriminate against a correctly behaving party [Aso98]. A simple exchange of signed messages is not fair in the sense that once the other party receives the signed message, that party can terminate the protocol, thus leaving the initiator without any evidence of the transaction. Thus, in order to be fair, at the end of the protocol either both or neither of the parties should be in possession of expected items. Note that this might seem similar to the two generals problem used to illustrate coordination challenges in distributed communication [Gra78], but the difference is that whether the parties are also able to assure an external party about the course of events rather than just assuring themselves that the other party has indeed received the message.

While fairness could also be implemented between just two parties using, for example, gradual exchange of a secret, such as in [Blu83], those mechanisms generally are not

considered very practical [Zho97a]. For instance, the exchange of a message might assume similar computing powers or require as many messages as the length of the message in bits, which is quite inefficient [Kre02]. Thus, for practical systems, one needs to consider the involvement of TTPs, as described above.

Fairness can also be defined to have different levels of strongness. Asokan defines strong and weak fairness based on the characteristics of the protocol [Aso98]. Strong fairness is implemented wholly within the protocol, thus the correct run of the protocol ensures that the parties will receive the needed evidence or, if not, have not committed themselves to anything. Weak fairness, on the other hand, may require that in the case of a misbehaving party, say `B`, the other party, say `A`, has to contact an arbitrator, a third party, and show that `B` has received an item from `A`, but `A` has received nothing. Thus, `A` can receive the missing item from the third party or some other actions, for instance, legal actions can be taken against `B`. [Gär99] defines an additional level between the two: The eventually strong fairness depends on additional assumptions about the participants, so that some state can be eventually imposed on a participant. A trusted computing platform might be one such assumption. [Kre02] defines true fairness, which in addition to strong fairness expects that the evidence is not dependant on the protocol execution. For instance, when looking at the evidence, one cannot tell whether the TTP participated in evidence generation or not. Thus, one could see this as a method of not casting undue doubt on a party, when the original evidence was lost, e.g., due to faulty network connections and the similar information was forced to be fetched from the TTP. [Kre02] additionally considers probabilistic fairness, in which the fairness of the outcome is based on probabilities. Naturally, one may question what the validity of such a solution would be when contested in court, for example.

One additional aspect of fairness is the timeliness. It relates to the proper and fair termination of the protocol and the possibility to do this in finite time [Aso98]. For instance, the protocol in [Zho96a] expects the participants to fetch the final confirmations of the exchange from the TTP at the end of the protocol. From a practical viewpoint, it is not clear how long these confirmations should then be available. As [Kre02] states, the protocols should not put a participant in a situation where the protocol session is open for an indefinite amount of time. In other words, the termination point of the protocol is unclear. This has a clear relationship to DoS considerations as well, because the invested communication resources should be released within a reasonable amount of time. Timeliness can also be related to liveness, which states that a programme eventually enters a desirable state, that is, something "good" will happen [Owi82]. However, it could be stated that the fairness also includes the safety property, that is, something "bad" will not happen [Owi82]. This emphasises the fact that even though the full exchange has not taken place, i.e., it was terminated prematurely, neither party has the upper hand in terms of possessing evidence.

4.3 Evidence and accounting

The previous discussion mentions the need to have evidence to prove the participation in the transaction and ensure the fairness of it. This evidence has to be such that, along with undeniable validity, the origin and integrity of it is verifiable by a third party [Zho97b]. Thus, evidence is the central piece of non-repudiation. In fact, it can be stated that the very goal of non-repudiation is to collect, maintain, make available, and validate irrefutable evidence, which can also be seen from the different stages in the process of non-repudiation [Her95]:

- Evidence generation
- Evidence transfer, storage, and retrieval
- Evidence verification
- Dispute resolution or arbitration

Evidence is generated along the protocol run. A piece of data signed with a private key of a user can be seen as generation of evidence of origin and integrity. Alternatively, there could be a TTP, which would generate the evidence on behalf of the user. This does not necessarily need to involve public key cryptography, but could be done with a secret key known only to the third party, in other words, a secure envelope would be created [Zho97b]. Once the evidence is generated, the accountability, that is, the association between the originator and the object or actions [Kai96], can be established.

Evidence needs to be transferred between different parties and stored accordingly. This could also be referred to as accounting, even though it has a general interpretation of resource consumption tracking and collection without requirement for non-repudiation per se [Abo00]. Storage of evidence brings forth traditional IT problems. For instance, how to make sure that evidence is not lost due to node crash or failure. Also, there might be some legal requirements as to how long to store the data.

Evidence needs to be verified in order to gain certainty that the received piece of data can indeed act as evidence. Thus, this will provide the confidence that the data in question can be presented in dispute resolution. A verification step can also be expected to check whether the evidence has been generated at the time of occurrence, i.e., it has proper timestamps in it. Generally, one could use a notary to produce timestamps, but as [Lou00] observes, this kind of timestamp still only tells that the notarised message was transmitted after the said time, not necessarily at that time. So, for instance, delaying the sending of a message that has an expiration time might leave the receiving party in an unfair position when trying to present the message to a third party, because the third party might consider the message expired. Revocation of keys can also affect the validity of evidence, e.g., whether a signing key was valid at the time of signature. It is worth

noting that an unscrupulous party might even try to revoke his own keys at a suitable instant in order to be able to make a claim that the evidence is not valid. [Zho99b] suggests using long and short term keys, where only long term ones are revocable and transaction signing is delegated to short term keys. The reasoning here is that the risk of having compromised keys is manageable within a short timeframe. Also, the extra burden of making revocation checks on high volume transactions may outweigh the benefits, especially when the transactions have low risk and value.

Dispute resolution takes care of the conflicting view the participants might have about the run of the protocol. This requires a third party, an adjudicator or a judge, which will assess the presented evidence and decide if a party has not been honouring the protocol. While the evidence might be technical, the process might be non-technical, that is, it might involve legal procedures. However, it is another thing whether the legal system conforms to the technical expectations and is able to digest the presented information [And94] or even trust the presented evidence [Bra10]. One can also ask whether the legal system believes the trusted party.

As mentioned above, accounting is part of the evidence handling. There are various accounting protocols which have been developed to keep track of the consumed resources and also to provide authentication and authorisation properties, i.e., to provide the "three A’s". RADIUS [Rig00] is perhaps the most widely used AAA protocol and provides good toolkit support, whereas its successor, Diameter [Cal03], has been enjoying success mainly in the cellular core networks, especially in the 3rd Generation Partnership Project (3GPP) context. It would also be possible to employ Common Open Policy Service (COPS) [Dur00] and Simple Network Management Protocol (SNMP) [Har02] to a certain degree [Mit01], even though originally they are not meant for authenticating end users.

As accounting protocols can be used to transport the reports of the resource consumption, they can also be used to transport the actual evidence related to this consumption. It naturally has to retain the accountability property of the evidence, so that it can be traced to the correct entity. From a practical point of view, the evidence should be easily transportable. For instance, relaying information about every packet is not really feasible. On the other hand, if the evidence is created just once per transaction, it leaves a larger timeframe for misbehaviour. It is better to try to have a granular approach, which provides incremental evidence over a certain interval, which could be time or volume based. Hash chains [Lam81] are especially suited for this kind of incremental approach, because at the time of accounting, one only needs to transport the first and last value of the chain and the amount of values can be calculated from those two. The idea behind hash chains is depicted in Figure 4-1. Additionally, they are computationally easier to verify, compared to public cryptography signatures.

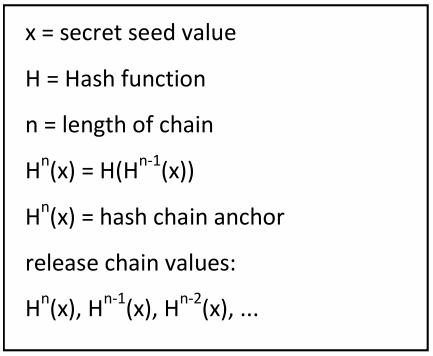


Figure 4-1. Creation of a hash chain from a secret seed value using a hash function H

4.4 Non-repudiation protocol examples

An often cited fair non-repudiation protocol is the one proposed by Zhou and Gollmann [Zho96a], depicted in Figure 4-2. In it the parties commit themselves to the transaction by exchanging signed receipts, which do not yet reveal the messages themselves. In essence, this means that the encrypted version of the message is exchanged. The fairness is ensured with the help of a third party, which will provide the encryption key to the message. Thus, in case the originator has released the key to the TTP, the recipient is able to access the message. In case of incomplete transaction, the recipient is just committed to the encrypted message and the TTP notes that the key has not been submitted. However, as pointed out by [Lou00], this does not take into account the practical aspects, such as how long the key should be available and whether the participants can actually receive the final messages in case of network failures. So, even though a protocol might look nice from the theoretical perspective and assume things like reliable transport, it is another thing whether it is practical to implement it. To be fair, [Zho96a] also suggests modifying the original protocol by including time limits to the protocol, but this can introduce new problems [Lou00]. TTP is, for instance, able to read the messages, because it still retains the possession of the key.

As the discussions in the previous sections have suggested, the TTP is central to the practical, fair non-repudiation systems. However, for performance reasons, one should carefully consider the involvement of TTP in the actual protocol. In this respect, the exchange could be optimistic, as suggested by [Aso00], so that the TTP is only contacted if something goes wrong and additional steps are executed to resolve the disputes or abort the protocol run. Zhou and Gollmann have also suggested an enhanced version of their protocol to reduce the involvement of the TTP to such that it is only needed, if it suspected that something went wrong [Zho97a], although it still might be problematic to execute the conflict resolution with the TTP in time, as noted by [Lou00]. Unfairness could also be introduced by running several kinds of similar exchanges, if the separation

of different exchanges is not sufficient, that is, the message receipt from a previous exchange can be applied to a later one [Gur03].

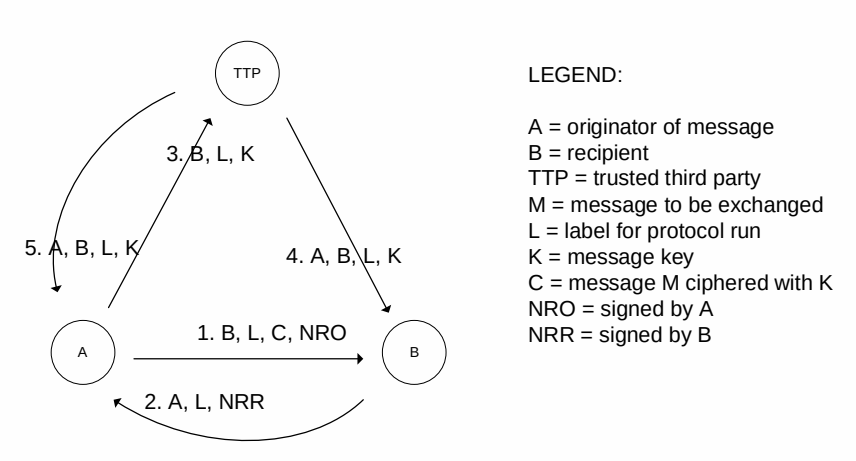


Figure 4-2. A fair non-repudiation protocol by Zhou and Gollmann for ensuring the exchange of a message and the corresponding acknowledgement [Zho96a]

Another aspect is the nature of data, which is to be non-repudiable. For instance, the above protocols basically consider individual messages or goods. On the other hand, the parties might be trying to sign a contract, for which they first ought to agree on the terms of the contract and then sign the actual contract itself [Rui06]. The previous approaches do not, however, really consider packet based data. After all, it would be quite inefficient to run the non-repudiation protocol for each packet, as noted earlier. Thus, one also needs to consider the granularity aspects of the data. Naturally, on the other end, previously mentioned gradual exchange of secrets, where bit by bit communication is already too granular for most applications.

A step in this direction is taken in the approach suggested in [Zho98], where the authors propose a system to provide non-repudiation of the calls a user makes in a GSM environment. It takes advantage of hash chains to produce evidence in a granular fashion, that is, the continuous submission of hash chain values is tied to the length of the service usage, much like in [P3]. It also involves a registration step with the home network, which is responsible for authorising the user, based on the shared secret between the home and visited networks. The same authors also consider the hash chain approach for web based video service [Zho99a].

A similar hash chain based approach has also been suggested for WLANs [Bla03], which also points out the importance of considering risk management alongside fraud prevention. In a sense, this kind of approach can be viewed as micropayment and similarities to the Payword concept presented in [Riv96] can be observed. They also argue that losing "a nickel" is acceptable for an ordinary user. [Tse04] also proposed to use hash chains to enable secure billing in a wireless environment, but that work concentrates on the authentication of the service request and does not really provide granularity

beyond a service request. [Im08] suggested hash chains for undeniable billing in a wireless environment, but as the scheme requires always connecting the home network and a separate endorsement entity for billing commitments with which the user shares a secret, it requires multiple roundtrips and does not provide true non-repudiation, because the endorsement entity could create billing records on behalf of the user. [Li09] targets GSM with a similar approach and has only a single entity in the home domain, but as it relies on a shared secret, it could also lead to situations, where evidence was not really generated by the user. This is an extra complication, when the evidence is presented to an external adjudicator and it has to trust the "trusted" third party as well. So, one should note the difference between the role of the TTP as an evidence generator or as the ensurer of fairness.

[Tew03] presents a micropayment solution employing hash chains to be applied for ad-hoc network packet routing, although it requires preliminary endorsement for the payment tokens from a broker and the provision of smart cards for the nodes in order to avoid double spending. It also suggests the possibility of using hash chain trees instead of individual hash chains. This way it is possible to save on the storage requirements at the expense of computing power, that is, multiple chains can be created from a single stored anchor value. However, in order to be energy efficient and considering the availability of cheap storage, nowadays in most mobile platforms one wants to reduce the amount of computations in order to extend battery life. One should remember, though, that often wireless communication is the most significant contributor to energy consumption [Rag04].

4.5 Practical aspects

While there are many protocols which suggest various ways to ensure the non-repudiation, like the ones briefly mentioned above, it is another thing how practical they are in the real networking environment. One aspect might be the need to have assured transport, that is, no packets get lost. One could argue that TCP will do just this, but some applications might be working over UDP or devising their own solution over IP. Thus, in some cases the loss of packets could cast a shadow of doubt over the honesty of the parties. For instance, in the above case, where the keys or evidence are retrieved from the TTP within a certain deadline, fairness would suffer from deliberately induced network failures.

Perhaps a more profound question relates to the security of the communication. Are there really guarantees about the authenticity of the parties? For instance, the participants might be named with such strings as A and B. This does not yet tell much about how these relate to real world entities, which have the real capability to pay their bills. One could use signatures based on public key based identities, but still one needs to have some sort of binding to the real entities. This is where the trusted third parties, such as telecom operators, usually enter the picture. The confidentiality of the commu-

nication might also be needed, for instance, to hide the nature of goods or the amount of payment exchanged. Thus, the users might wish to preserve their privacy and prevent others from learning their purchasing habits or what sort of content they access. A typical response might be that IPsec will solve the issue, but this really does not say how it should be done. IPsec does not just "happen", it requires configuration steps, in which proper keys are exchanged and the trustworthiness of the parties is ensured, i.e., the aforementioned authenticity problem.

[P3] presents how this problem space can be approached by integrating the non-repudiation solution to a key exchange and identity establishment protocol, i.e., HIP. The approach suggested in [P3] also takes into account the possibility to have privacy protection, at least towards the external observers. This basically requires delegation to an ephemeral identity, which is responsible for representing the party of the transaction. In a sense, this has similarities to the proposal mentioned above, which suggests employing long term revocable and short term irrevocable certificates, even though the motivation for this proposal was evidence validity rather than privacy [Zho99b]. In such mutual communication involving real world effects, like charging, the other party ought to gain certainty about the trustworthiness of the used identity or that there is a party who will pay the bills. If the TTP were involved in the transaction, it might be enough to provide suitable assurance. However, it depends on the nature of the transaction as to whether one wants to contact the TTP every time. It is partly a question of performance due to added roundtrips, but it also may not be in the interest of the TTP to be contacted frequently, because it is likely to make it more expensive to run the TTP functionalities and the TTP could easily become a bottleneck. On the other hand, it might also be so that the TTP is actually a party, which provides identity management services and charges for each transaction, such as authentication. Thus, it would be preferable that the TTP is contacted as few times as possible and most of the interaction takes place just between the parties, as suggested by [P3].

Typically, at the time of initial network attachment, the service provider might have the strongest incentive to contact the TTP to get real-time confirmation of the validity status of the user. This kind of approach is described in [P4], which also provides a practical way of transporting the evidence of the service usage to the TTP for clearing. Similar ideas were also described in [Hei05], which introduces additional roundtrips and does not consider the practical aspects of accounting exchange that extensively, but provides the home network endorsement to various kinds of services with different kinds of charging policies. Both proposals then rely on the hash chains approach to provide the granular service usage solution. The former proposal, however, is more flexible in its approach as there is no need to contact the home operator every time. After all, one could expect that the majority of the transactions are honest, especially in high volume services, so the overall burden is smaller, when only certain transactions, based on the risk management policies, are deemed to require stronger assurance. The former pro-

posal also allows one to take advantage of RADIUS roaming infrastructures, when the user name is presented, for example, with the format HIT@operator.com.

The online consent from the TTP also relates to the revocation concerns. This is especially relevant when it comes to issues like key compromise. The suggested method of having short lived authorisations from the TTP is not fool-proof, but it has performance benefits over online methods, which require contacting a separate revocation service every time. It is worth noting that it is basically up to the service, as to what degree one needs liability guarantees. After all, the employed identities are self-certifying in the sense that authenticity and the origin can be checked by relying on the cryptographic properties of the identities themselves, without contacting any external infrastructure.

The above consideration also leads to the question of whether there are enough incentives for different parties to deploy a system, which provides non-repudiation. From the TTP or home operator perspective it is a question of how much more additional investments are needed and what the added benefit is. While from the business point of view it is possible to make a decision that the risks are already at an acceptable level and the current risk management techniques are sufficient, one might ask whether this is the case when the dynamism in network interaction increases. This dynamism relates to the ambient networking setting, which suggests that even small players can assume the role of the operator and interact with other operators in the same way as the current incumbent operators interact with each other through static arrangements [Nie05]. It is likely that incumbent operators do not wish to embrace this kind of development, which would mean increased competition. Thus, regulatory actions would be needed [Mar07]. If we assume that this kind of development takes place, then there is greater incentive for the operators to also protect themselves and their customers from potential rogue operators. Likewise, the smaller operators want to be in possession of concrete evidence of their service provisioning, in case the users try to avoid their bills by making claims of unauthorised charges. Such claims can exist even now in the credit card industry, when so called charge-back mechanisms are used and the customer denies ever having made the credit card transaction [Del09].

From the migration point of view, naturally it is always challenging to introduce new functionality. Thus, the solution should be easily deployable. If the solution is working on higher application layers and over existing platforms, it is easier to deploy, for instance, by relying on a combination of HTTP and TLS. However, a more general solution would work on lower layers. If the future networks were to embrace HIP as a standard solution, then the proposal in [P4] would provide quite a natural approach. Migration to HIP provides its own challenges though, as it touches the network stack and its widespread adoption is still years away.

4.6 Identity based approach for non-repudiation and accounting

The identity based approach allows us to consider the whole non-repudiation scenario from a holistic perspective, so that we are not just considering the exchange of undeniable evidence, but also how it relates to real communication protocols and the entities involved in the transaction. HIP provides a natural communication framework for identity based accounting solutions, even though there are also other major benefits compared to other approaches relating to mobility enhancements and DoS mitigation. After all, one could envisage devising similar solutions, for instance, by employing modified versions of protocols such as IKEv2 [Kau10] and TLS [Die08]. IKEv2 with IPsec would result in a similar kind of solution, and even though TLS, being on a higher layer, would allow more natural session separation, traffic selectors in IKEv2 can be used to differentiate between different kinds of traffic flows.

Introduction of non-repudiation properties to TLS would basically require the definition of a new subprotocol for the negotiation step and the signalling subprotocol for exchanging the evidence, that is, protocols running on top of TLS record protocol. One could, of course, instead modify the basic handshake protocol to include those functionalities in order to save roundtrips, but from the modularity perspective it would be better to define new subprotocols. Also, even though [Kau10] defines an extension mechanism for TLS, it is not as flexible as the one with, e.g., HIP. It is worth noting that this kind of approach also deviates from the most common way of using TLS, in other words, nowadays basically only the server is providing a certificate and the exchanged messages do not use signatures. Hence, the modified protocol ought to ensure that there is a binding between the used identities in the non-repudiation negotiation and the traffic.

In IKEv2 the choices are basically the same: integrate with an initial handshake or devise an extension approach. The latter approach is more straightforward as one could define a new child Security Association (SA) for non-repudiation functionality negotiation and use a notification mechanism to transport the evidence tokens. It would also decouple signalling and data traffic more clearly than the TLS alternative. It is worth noting, though, that this contradicts one of the original design criteria of IKEv2, i.e., plausible deniability [Hof02]. In other words, a party is supposed to be able to deny its involvement in an exchange with another party.

From the migration and deployment points of view, the previous two are actually better choices, because, as mentioned, HIP is not a widespread solution as of yet. However, it is still worth remembering that interoperability with non-supporting implementations could suffer, if the legacy ones could not handle well the new parameter and payload types defined for TLS and IKEv2. Other application layer solutions could be devised as well, but they would be application specific.

As already discussed in the previous chapter, there are also some other research initiatives to introduce stronger accountability to the inter-networking level, such as PLA, CGA, and AIP. They, however, concentrate more on the accountability and authenticity of origin aspects of IP packets rather than accounting and usage reporting. Thus, the mechanisms are not so usable when it comes to presenting evidence to a third party. In addition, AIP suggests a disruptive architectural change to the Internet addressing by using flat, self-certifying identifiers as addresses. CGA and PLA, on the other hand, could cooperate with the HIP based approach in IPv6 networks. For instance, [Lag10] suggests such coexistence of PLA and HIP. There also exists a HIP based proposal in [Hee09], which suggests including hash chain tokens to IPsec packets to enable per-packet authorisation for the benefit of middleboxes.

Thus, HIP can be made middlebox friendly in order to allow in-path devices to act upon the traffic. For instance, when a firewall notices that there is no more payment for the service, it can then close the corresponding ports. In addition, the schemes described in [P3] and [P4] demonstrate how the different parts of the service interaction can be strongly bound to the used identities. Firstly, the identities are derived from the public keys of the participants. Secondly, the handshake procedure authenticates the identities and the integrated negotiation step messages are signed with these respective identities. The hash anchor is also part of the signed data, thus the stream of evidence tokens is bound to the identity in question. Thus, when a customer submits a hash chain token as evidence at agreed intervals, there is a certainty about the identity, to which the evidence is bound. In addition, as the handshake derives session keys, the client can trace the protected service traffic to the correct service provider.

However, it should be evident that this approach does not guarantee strong fairness. Strictly speaking, it is not even weak fairness as it is not completely symmetric, when it comes to gaining evidence. The service provider has the advantage, because he has the evidence (tokens), which can be presented to an adjudicator (which we assume to be the home operator of the service consumer) in order to ascertain correct payment. The customer does not have such evidence, unless he wants to record all the traffic exchanged with the service provider. However, if it were desired, such evidence could be included in the acknowledgement packets, which would transfer hash chain values separately bound to the service provider. For a typical use case this might not be needed, but one could employ this kind of setting to prove that a certain amount of service has been used, for instance, to show that a commercial has been viewed, which in turn could function as “payment” for some other service usage.

It is worth noting, though, that the scheme still allows dispute of at least one unit. In any case, we rather say our scheme implements accounting fairness, because the customer validates the correctness of the service as it is used. In other words, as long as the service is provided correctly, the customer knows that he is accounted fairly. Naturally, this approach is mainly suited to services, which can be easily sliced into small units. As

we are only interested in creating accounting evidence instead of more complex issues of micropayments, we skip the problems of double spending and unauthorised coin generation. In this respect, one can consider the exchanged items in our approach to be idempotent, that is, possessing the item once does not differ from receiving the item multiple times [Aso98]. Of course, if the service provider can break the hash function, he is then able to generate additional evidence tokens, at least to the end of the chain. However, if the hash function is assumed to be secure and the value of a single token is expected to be small, then this is not really feasible to do.

5. IP MULTIMEDIA SUBSYSTEM

5.1 Introduction

While traditionally the telecom networks have concentrated on providing voice based services (calls), the proliferation of the Internet has shown the value of a wider array of rich services. Thus, the need for accessing these services, irrespective of time and place, has increased considerably and the wireless Internet is blooming. This has resulted in discussions about the convergence of different access methods, so that services could be provided in a seamless fashion. However, a tussle between the operators and service providers has emerged, since the operators are not content in just serving as bit pipes, but instead would like to get their share of high margin services. This is not entirely without justification as the service architectures have been quite diverse and especially flexible charging, traditionally a strong asset of the operators, has been problematic. From the other perspective, the operators have not been that forthcoming in relinquishing their control of access services. In other words, the access services are mainly offered by large operators which interact in static and inflexible ways.

In this chapter, we take a look at one service architecture, IP Multimedia Subsystem (IMS), and discuss how it positions itself in the above context. It has been argued that it is just a desperate attempt of the operators to bring their "walled gardens" into the service provisioning, but we try to investigate whether some enhancements to the contrary could be provided with the help of identity based approaches. This is especially relevant in the evolved ubiquitous networking environment where the relationships are expected to be more dynamic in contrast to the current static operator agreements.

5.2 Benefits of IMS

A common question regarding IMS is why one would need it, because basically one could use the Internet services in the wireless domain just using packet based access, which is already readily available. [Cam06] suggests that the answer lies in three factors: quality of service (QoS), charging, and integration of services. As typical Internet services are provided with the best effort quality, the high QoS is seen as an essential requirement for ensuring a good service experience, especially when dealing with a resource constrained environment and services with real time requirements, such as voice. However, one should also remember that the end user environment, like radio noise, cannot be completely controlled by the infrastructure and the end-to-end path may not

be in control of the same entity. Also, in core networks operators have seen overprovisioning as a viable option to ensure the availability of bandwidth [Pra05]. One should not overlook the value of transcoding though to adapt, for example, media streams according to the capabilities of the terminals in order to save resources.

Charging is given as an incentive to be able to bill different services in a flexible way. It is true that the operators have been good in devising architectures to ensure billing even of small amounts, which is currently challenging in the Internet on the large scale. Customers, however, have gotten used to the flat rate billing models of their Internet access and the suggestions that threaten net neutrality have been met with uproar. The operators are trying though to move back to models, which also take into account the usage, e.g., in the form of data packages. A cynic might claim that this is due to their unwillingness to invest in the infrastructure. Such economic feasibility discussions are outside the scope of this work, but we note that dynamic cooperation, such as proposed by the Ambient Networks concept, can be an instrument for decreasing investment costs [Nie07].

Perhaps the most interesting thing about IMS is the integration of the services aspect. It gives the possibility, at least in theory, to create services flexibly, without having to standardise them all like has been the traditional telecom approach. After all, interoperability has been an important factor between the operators. Thus, different kinds of services can be combined to create richer services, in the similar fashion of the Service Oriented Architecture (SOA) concept [Pap03]. Naturally, the key role of the IMS operator as the controller of service discovery and orchestration can still cause anxiety to the most vigorous proponents of Internet freedom. Still, with the advent of Long Term Evolution (LTE) and the selection of IMS as the vehicle for voice and messaging [Low10], the significance of IMS is bound to increase in the future.

5.3 Architecture

From the IMS perspective, the communication system can be seen to be separated into three different layers or planes [Tar07]. The connectivity layer gives the basic communication capabilities, or transfer of IP packets, and IMS itself intends to be access agnostic. The core functionalities of IMS take care of the next layer, i.e., control, and it includes the signalling exchange between the logical elements with the help of Session Initiation Protocol (SIP) to enable the various services and provide the session management capabilities. In the service layer the various services and applications, such as presence, interact with the users using whatever transport services were negotiated. Thus, IMS is the enabler of services.

An overview of the IMS architecture is given in Figure 5-1 (adapted from [Cam06]). It describes the entities and their relationships, even though not all of the elements are shown for the sake of simplicity. For instance, different media and transcoder elements

are left out. There are also elements responsible for charging and billing functions and they have connections to most of the other elements as well [32.260]. It is worth pointing out that even though the figure might show different nodes, they are actually functions. Thus, IMS is about standardising functions instead of nodes, and the multiple functions could be co-located in a single physical node [Cam06].

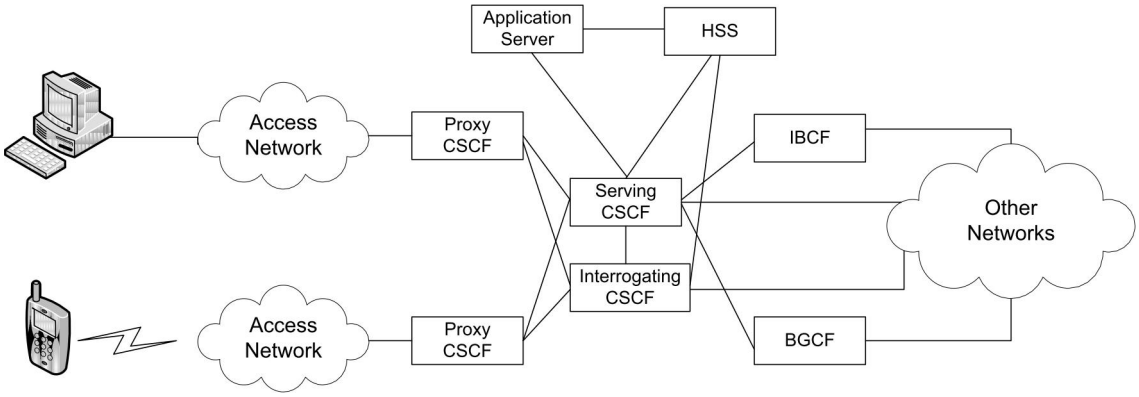


Figure 5-1. Simplified IMS architecture showing the most important logical parts and their connections (adapted from [Cam06])

The architecture consists of various Call Session Control Function (CSCF) elements, which are in essence SIP proxies and control the signalling flows. There are also user databases, most notably the Home Subscriber Server (HSS), which is responsible for storing subscriber related data, such as user profiles and authentication data. Some functions are responsible for controlling access to other networks, such as the Breakout Gateway Control Function (BGCF) and Interconnection Border Control Function (IBCF), although their setup is also dependant on the policy and topology choices of the operator (such as, whether to introduce topology hiding [33.203]). From the application point of view, Application Servers (AS) are of great interest as they are responsible for implementing the service logic.

Proxy CSCF (P-CSCF) provides the entry point to the IMS functionalities for the terminal [23.228]. Thus, it is responsible for finding the next signalling contact point, which might be in the local or remote network. It also has to create a security association with the terminal, based upon the authentication procedure, to which it is a passive forwarder. However, after the user authentication, it is responsible for including the authenticated user identity to the SIP messages, that is, it asserts the user identity, even though it does so with no real integrity protection. P-CSCF also has the important functionality to interact with the connectivity layer and ensure that relevant resources for service implementation are available. This takes place in cooperation with Gateway GPRS Support Node (GGSN), which is responsible for network level packet routing.

Interrogating CSCF (I-CSCF) serves as an entry point to the core IMS network [23.228]. In some cases it can also work as an exit point, although IBCF has taken much of this role in newer specifications. The main role of I-CSCF is finding the correct

Serving CSCF (S-CSCF) to serve the user in question. This takes place with the help of HSS, which gives the address of the assigned S-CSCF.

Serving CSCF (S-CSCF) performs most of the session control services of the user [23.228]. It is basically a SIP registrar as defined in SIP RFC [Ros02a] and it handles the registration of the user to the IMS network and maintains the binding between the public identity and the location of the user. All the signalling messages of the user travel through S-CSCF and it has the knowledge of what to do with them. It could forward them towards other networks or could decide, based on the profiles that certain AS can take care of the associated service request. It is also possible that there is a chain of ASes, which the request traverses in order to fulfil the requirements of the service. From the security perspective, S-CSCF has the important role of deciding about the authenticity and authorisation of the user, based on the information received from HSS.

5.4 Identity issues

The user identification in the IMS is mainly done through the use of private and public identities [23.003]. The user has a private identity, preferably stored securely in a Universal Integrated Circuit Card (UICC), or smart card, even though strictly speaking it is contained in the IMS Subscriber Identity Module (ISIM) application within the card. The private identity is the one that uniquely identifies the user (or the subscription, rather) and gets authenticated during the registration process based on the shared secret available to the ISIM and the home network. The format follows Network Access Identifier (NAI) syntax, i.e., username@realm [Abo05]. The public identities of which the user can have one or several, get registered as active ones during the registration, and as SIP URIs are the ones that are used for message routing purposes, for example, sip:username@operator.com [23.228]. Thus, the IMS identities are basically human readable application level identities.

5.5 Connectivity attachment

Before the user and the accompanying terminal device are able to take advantage of the IMS services, a number of procedures are needed to execute the attachment. While there is the prerequisite of having a suitable contract, or subscription, one also needs to get IP connectivity, discover the address of P-CSCF, and register to the system [Cam06]. As IMS is supposed to be access agnostic, IP connectivity can be established in a multitude of ways. This includes, for instance, General Packet Radio Service (GPRS) access over 2G and 3G networks [23.060], 3GPP WLAN access [23.234], and evolved packet core access, i.e., LTE [23.401]. Currently, GPRS is the dominant packet access method in cellular networks and the connectivity is gained through the process called GPRS attach [23.060]. It starts with the request of the terminal for activating Packet Data Protocol (PDP) context. This is directed to the Serving GPRS Support Node (SGSN), residing in the serving network. Based on the request, it will send a PDP context creation request to

the Gateway GPRS Support Node (GGSN), which will then be responsible for assigning an IP address for the terminal. At the same time, GGSN can also give the address of P-CSCF, but it is also possible that DHCP is used at a later stage to request this information [Sch02][Sch03]. DHCP is also the most likely choice in other access bearers. Note that in the case of IPv6, it might be that only the prefix of the network is given. After the address of P-CSCF has been discovered, the user is able to register to the IMS system. That procedure completes the attachment and is described in the following section.

5.6 Session control

As indicated, the signalling in IMS is done mainly through the use of SIP [Ros02a], which as an application layer protocol can run over multiple transport protocols. It is basically a text based protocol, which borrows heavily from HTTP [Fie99] and SMTP [Kle08] and follows a similar request-response model. By working in a hop-by-hop fashion it tries to discover the correct session participants and negotiate the session parameters between them, which happens with the help of Session Description Protocol (SDP) [Han06] carried within the body section of the SIP messages. A signalling path is created when the SIP messages are routed through SIP proxies, but the actual session traffic may take a completely different route between the participants. Due to their fundamental nature, we briefly go through two specific signalling cases, that is, how registration and session setup are conducting in IMS using SIP. An interested reader can refer to [Cam06] and [24.229] for more detailed descriptions.

In order to be able to benefit from the services of an IMS system, the end user has to first register. It is also needed, so that the system is able to tell where to find the user. In other words, a binding between the user identity and the location (typically IP address and the port number of the terminal) is created [Ros02a]. In typical SIP, the registration takes place by sending a REGISTER message to the registrar. In IMS, this means sending the message to the home network, where S-CSCF acts as the receiving entity. The process starts by sending the message first to P-CSCF, which in turn finds the next hop in the signalling path, for example, using DNS. In case visited and home networks are different, typically this is I-CSCF (or IBCF) in the home domain. I-CSCF is then responsible for finding S-CSCF to serve this particular user and it uses the information received from HSS to make the decision. Naturally, at this point it is already verified whether the given user identity really is under the control of this home domain. After this, the correct S-CSCF is contacted. S-CSCF fetches the authentication vector from the HSS to initiate the authentication of the user. Thus, a challenge is issued and is forwarded along the same path as the initial registration message (SIP message headers are used to record the proxies, which wish to be on the signalling path). As a by-product of this, P-CSCF also receives keying information, which can be used to establish a security association with the user terminal. Once the challenge is received, the response is calculated in the corresponding ISIM application and sent back in another REGISTER mes-

sage. When it reaches S-CSCF again, it can verify the response and set up a state for this user, which includes the initial filter criteria for determining how to handle the forwarding of user messages for specific services, that is., which AS should receive them. S-CSCF finally sends a message back to the user and acknowledges the authentication. P-CSCF will also notice from this message that the user authentication was accepted.

In order to setup a session with another user, IMS, like in typical SIP, uses an invite procedure. Note though, that the invite mechanism could be used to signal access to some service as well. The procedure follows a similar path as in the case of registration initially, i.e., it travels towards the S-CSCF of the home domain. However, P-CSCF first does some processing as it is the entity responsible for making sure that only authentic invitations with suitable resource requests are sent, coming from channels with adequate security association. This basically means that P-CSCF includes in the message the identity of the user it considers to be the correctly registered one [Jen02]. However, no specific security measure is applied to this information. When the message reaches the home S-CSCF, it first checks whether the request is within the policies of the network and the subscription, and then it has to decide how to find the called party. In case it happens to be in another domain, it has to initiate similar procedures as P-CSCF had to initiate with the registration message to find the entry point of the destination network. Eventually, the assigned S-CSCF of the called user in the destination domain is contacted and it forwards the invitation towards the user, even though it might still travel to another P-CSCF in a foreign domain, if the called user happened to be roaming too. In case the called user answers, an OK message is sent to the terminal of the calling user, which then acknowledges this with an ACK message.

However, even on a high level this procedure looks like a three-way handshake, but there is actually much more signalling going on, which involves informing the different elements about the progress of the session [Ros02b] and negotiation of the session parameters (especially the status of QoS with preconditions [Cam02] in order to ensure that the relevant resources are reserved from the network, before the user is actually alerted). After the signalling and the possible extra resource reservations, the end entities are then able to directly exchange traffic, for example, the actual call. This takes place on the IP layer and traverses the IP layer networking elements. Note that this traversing depends on the location of GGSN (and P-CSCF, as they are assumed to be in the same domain), thus the operator might want to make sure that all the traffic of its subscribers goes via its domain for ensuring proper traffic charging. On the other hand, the operator might not be certain about the IMS capabilities of the visited network and therefore might want to ensure the proper operation using the network elements whose properties it knows [Cam06].

5.7 IMS security

The main security issues considered and specified in the IMS setting relate to the registration of the users, security association establishments between the user and P-CSCF, and securing the interaction between the core IMS element. The first two are tightly related as they take place during the same dialog and are considered to be part of the access security [33.203]. The third considers the network security aspects [33.210].

The register procedure was already discussed earlier from the session control point of view, but here we go through it again from the security mechanism point of view. To start the registration, the user indicates his private identity in the REGISTER message and it also indicates to P-CSCF the security properties it is able to support for their mutual security association. So, in essence, there is no explicit security at this level at this point. Based on the received private identity, S-CSCF is able to query from the HSS the authentication data related to the user. This takes the form of an authentication vector, elements of which allow S-CSCF to authenticate the user, the user to authenticate the home network, and keying of the security association establishment between P-CSCF and the user. The authentication information is similar to that used in the basic Authentication and Key Agreement (AKA) procedure [33.102]. Only the method of transferring this information between the elements is different, that is, following the mechanisms given in [Nie02]. When receiving the S-CSCF initiated challenge, P-CSCF forwards the message to the user, but strips the keying information from it. It also includes its own security properties, so that any further communication between the user and P-CSCF can be protected with the negotiated security measures. From the received message, the user is able to check the token for home network authentication. After this, the registration is concluded as explained earlier.

The network security part is based on the 3GPP thinking of security domains, which are defined to be networks that are under the administration of the same authority and typically the traffic is policed in a similar fashion [33.210]. Within the security domain the network elements can choose to protect their intra-domain connections, but it is not mandatory. On the other hand, it is mandatory to at least implement integrity protection across the security domain boundaries, i.e., between different operators. This traffic is handled by dedicated security gateways. However, as stated in the Annex of [33.210], the IMS expects these interfaces to be confidentiality protected as well. This is quite understandable, because, as discussed previously, the visited P-CSCF gets the keying material unprotected from the home network. This whole model is based on the assumptions that the operators keep their internal networks secure, thus the information coming from the other operators is trustworthy. A good example of this is the assertion made by the visited P-CSCF about the identity of the user, which is used as a basis of charging decisions. However, operator internal networks may not always be that secure, as portrayed in [Pre07].

It is also worth pointing out that the security is applied in a hop-by-hop fashion, hence end-to-end protection is not supported. Especially, end-to-end confidentiality would conflict with mechanisms for lawful interception [33.107], which is required in certain regions at least [EC96]. Also, it might interfere with QoS planning, like using transcoding. While the default mode of operation for setting up these security associations is through pre-shared secrets, there is also a specification which offers an optional authentication framework applying a public key based approach with cross-certification [33.310]. An interesting concept there is the suggested Bridge CA, which could allow the extension of the trust model towards a more dynamic setting, i.e., instead of mutual agreements, trust delegation could be used [33.310]. It is another thing though whether the legal implications are acceptable, if, for instance, such delegation could lead to an automatic creation of a contract. In addition, the decision to exchange traffic with someone could be motivated by business reasons as well. However, in terms of signalling traffic, the situation might not be so convoluted as usually there is a real need to accept this kind of traffic, that is, it relates to serving the operator's own customers.

One additional aspect of 3GPP security, the so called Generic Bootstrapping Architecture (GBA), is the authentication and key agreement support for applications, offered either by the operator or third parties [33.220]. This is quite a natural approach, as the existence of a protected and pluggable environment with a shared secret (for example, UICC with ISIM application) along with the accompanying AKA procedure can be seen as a valuable bootstrapping asset. With this process the user is able to authenticate with the home network and based on this procedure can generate keying material, which the home network also delivers to the application. Thus, the home network has to be able to trust the application, which is basically identified by its DNS name. One specific application scenario for this procedure is the possibility to distribute certificates to the users [33.221].

5.8 Enhancing IMS

Much of the aforementioned security in IMS relies on the fact that the interacting operators are trustworthy and give valid statements to each other. For instance, as already stated, after the authentication, the identity assurance is made by the visited P-CSCF with no added security, although transport on each hop could be secured with, for example, TLS. Similarly, it is not expected that any unauthorised entity is able to maliciously modify the message headers. There are, however, already mechanisms for SIP, which take measures to provide stronger security for these (in addition to mere authentication). One very basic mechanism is described in SIP RFC [Ros02a], which relies on the use of Secure/Multipurpose Internet Mail Extensions (S/MIME) bodies [Ram10]. In other words, header information can be duplicated in the protected part of the message body in order to ensure the integrity. Naturally, confidentiality services can be provided as well, as long as SIP functionality is ensured (one still needs to be able to identify sig-

nalling destinations and sessions). A signature mechanism covering certain critical headers is described in [Pet06]. However, the relevant extra headers to identify the entity and the signature are added by a trusted proxy instead of the end entity. Signature information could also be added to the existing headers [ElS08], even though the signature covers only a couple of header fields.

Additional approaches are also suggested, which employ, e.g., identity based cryptography [Rin06a] and certificateless public key cryptography [Wan08], even though they rely heavily on the existence of TTPs to manage master secrets. It is worth noting, though, that in an IMS environment there is profound involvement of the operator and the possibility to have key escrow works for lawful interception requirements. Various other proposals discuss the possibility to have assertions about the capabilities of the end entities [Tsc06][Gai07][Mar03]. It is also quite natural to talk about entity certificates in relation to TLS, but in SIP the interaction is on a hop-by-hop basis, thus defeating the feasibility of using TLS in an end-to-end fashion (although semantics of SIPS URI expect every leg of the path to be protected [Aud09]). In the official standard track, there also exists a separate mechanism for retrieving entity certificates [Jen11].

Similar ideas can be applied in an IMS setting to enhance the security properties of the system, and from the service architecture perspective additional benefits can be gained, if the actions can be bound more tightly to the end entities, that is, assure accountability. This also includes delegation aspects, so that the signalling can be made more efficient with separate authorisations, and the possibility to refer to external entities in a secure fashion [Kop05]. With the application of suitable identities with security properties, it is possible to provide a holistic approach, so that synergy benefits for security can be gained in different parts of the system. Such a process binds the access level operations more tightly to the operations executed on the application level. In other words, the same identities are used to ensure the security of the network attachment as well as to assure the identification of the entity on application level signalling. With this approach, it is easier to ensure the identification of all the parties and the existence of the same entity in different parts of the system. This kind of system is described in [P5]. It firstly suggests using a common attachment procedure between the user and the access operator, for example, in the fashion of [Hei05], which allows creation of identity association and protection of the first leg of communication. The actual user registration to the home network is described to be like the normal IMS registration, even though one might also employ the public keys of the user and the home network for devising an authentication method. This would not leverage the existing investments to the operator authentication infrastructure so much, though.

In any case, before the registration takes place, the visited and home networks need to establish a roaming agreement if such an agreement is not already present (that is the assumption in the proposal). The user identity can work as an incentive and through it one is able to assure that there is a legitimate need for establishing the agreement. In

essence, the user authorises the visited network to provide the connectivity service. It works both ways because the home operator is able to provide its own authorisation after the agreement. Thus, the user can also note that the visited network is the same one that is providing connectivity and has established a roaming agreement with the home network. The proposal also describes the possibility to employ a hash chain based SIP level accounting solution for granular service usage, much like already described for the network level services in [P3].

One can take the identity establishment a step further and do it between the end entities as well. While solutions like Multimedia Internet KEYing (MIKEY) [Ark04a] could be used, one can also adopt an approach closer to the host identity ideas, such as the one depicted in [P6]. It takes advantage of IMS signalling as a transportation mechanism to convey HIP signalling between the parties. Thus, it leverages the available trust relationships and the compatibility with the HIP processing capabilities within the end entities. This can further be extended to service usage scenarios, where the users are the service providers [Hei09]. It could be argued that this is exactly what IMS is intended to be, that is, a service provisioning platform. However, IMS does not provide mechanisms, beside standard logging, which could be used as evidence if something goes wrong (or that something took place), so the motivation for allowing other than purely operator provided services is decreased. Additional motivation for an operator to allow this approach is that the operator is still able to access the signalling information and the related evidence. Thus, it can keep track of the service usage and provide QoS suitable for the service in question. One could note though that the colluding users could bypass the operator and just exchange traffic directly, but then the user providing the service would not be likely to receive any compensation, e.g., payment. After all, the operator is in a good position to do the clearing procedures needed to exchange charging information with other operators on behalf of the user. The scheme presented in [Hei09] does not, however, consider the content of the services per se, even though one could envisage services like streaming of video from some event. It is another thing, whether the provider would be authorised to provide the service, for example, there might be intellectual property issues. Such issues are evident already today with user created content, and similar litigation procedures against the hosting providers could be initiated as the providers can be identified. However, such litigation could extend to the operators as well, thus mitigating the incentives to function as a service discovery platform for unvalidated service offerings, unless some liability shifting can be done. This has already taken place though in some mobile phone application stores. There needs to be systems for tracking unauthorised provisioning and incident management though.

In the above the issue of employing identities in a holistic fashion was already touched on. One can still question the use of network level identities on an application level. However, one could envisage that HIP identities can be created from user provided identities as well [Kom05]. For instance, they could be residing in a pluggable authentication device. Thus, UICC could be hosting a suitable key, which could be employed

with the proposed identity based approaches. Naturally, this can allow key escrow for the UICC provider, hence mitigating the non-repudiation characteristics. On the other hand, a trusted platform that is harder to compromise is a more valuable asset.

5.9 Towards ubiquitous service communities

Network convergence has been a popular buzz word for quite some time. It relates to the heterogeneity of the existing access networks and devices and the suggestion that the users should be able to be connected irrespective of time and place [Nie07][Gus03]. Often IP is seen as the foremost enabler of interconnectivity, but due to its traditional fixed approach it is not always able to provide a seamless usage experience. There are various mobility technologies to help in this though. However, the users are mostly interested in services and the added value the connectivity can provide them. Thus, it is also important to consider this from the perspective of service platforms, like the above described IMS, and whether they are able to provide seamless service experience. IMS can be, after all, seen as a horizontal service enabler, that is, applications are able to take advantage of the common building blocks to implement the actual service. Even though this can be seen as very telecom operator centric, it still deviates from their traditional vertical service silo model [Räi05].

Even though IMS could be seen as a tightly connected and operator controlled infrastructure, it still has the possibility to interact with various other domains. The "wild Internet" can still provide SIP enabled applications, with which it is possible to interact, even though from the policy perspective the operators might be wary of this as these can compete with their core services, e.g., voice calls. There also have been research efforts, which suggest a more distributed approach for the signalling, that is, Peer to Peer SIP (P2PSIP) [Sin05], which can be made to interact with IMS through gateways [Hau08]. This line of thinking has also lead to suggestions of making the basic architecture of IMS more along these lines, that is, distribute the functionality to peers of equal status [Mat07]. While this approach might provide better scalability and robustness, it also requires a more extensive risk assessment as the capabilities and the responsibilities of individual nodes are higher.

From individual services one can move towards a service portfolio, which meets the needs of a certain user set. In a sense, this can be seen as creating a community of its own. [P7] presents an architecture which develops this idea based on a trusted community concept, where the community is initially defined by its services. It discusses how the different actors can get assurance about the authenticity of the other entities involved in the community. In other words, the users are able to identify each other as the members of the same community, but they are also able to verify the authenticity of the community provided or endorsed services. This is based on the existence of secure identifiers and the possibility of linking them together. While this allows taking advantage of the host identity concept by defining it to be part of the community, it is also possible

to have other community specific identities to express the membership and usage privileges. Similar domain concepts were already suggested within the Ambient Networks project, even though it basically considered the node membership within the network domain and the access to the resources the node in question would be willing to give to other entities, both internal and external to the domain [Nie07]. There have been other suggestions for using public key certificates to indicate community membership, for example, [Pea02], and, in fact, PGP could be seen as a community based approach as well [Zim09]. Communities can also be defined around collaborative business contracts accompanied with suitable trust and reputation management systems for establishing and monitoring the contracts [Kut07]. Research on secure communities can be seen as one potential future research topic.

While membership (or subscription) authenticity is important, it also has to be transferred to the interaction of the domains and communities. In the operator world, the earlier mentioned roaming agreement is a central concept in ensuring that the different domains can talk to each other and exchange traffic. Typically the approach has been quite a static one, but there is an identified need to be able to have more dynamic mechanisms as well [Kap07]. This is especially true in the ubiquitous visions for ensuring seamless access everywhere, that is, everybody should have the chance to initiate collaboration with everybody, even with previously unknown entities [Sei04]. While this could be possible from the technical side, from the policy side one needs certainties that there is a reason to engage in such interaction. As portrayed in [P5], a statement with relevant assurance and identification properties given by a roaming subscriber can function as such. Such decisions are still based on risk management actions, that is, evaluation between the estimated risk and the tolerated risk (uncertainty vs. potential benefits) [Ruo10]. Thus, such trust management mechanisms should be installed. For those, one is able to take advantage of the ownership of the identifiers to, for example, instil reputation scores (perhaps even exchanged through some federation) to the said identifiers. This makes defaming harder, but, like in most reputation based systems, previously well-behaving entities can still misbehave.

However, it is still questionable from the policy perspective, whether the incumbent operators are willing to cooperate with the local small operators. It might provide the big operators the chance to save on investments, especially when it comes to new, short range technologies, but also works as a disincentive to invest in core infrastructure for players seeking to take advantage of the existing infrastructure and just concentrate on high margin services [Nie07]. To solve these tussles, one might eventually need regulator intervention [Mar07].

6. CONCLUSIONS

This thesis has investigated the feasibility of host identities in securing various kinds of communication. The emphasis has been on the attachment procedures, both from the initial network attachment and the subsequent service attachment perspectives. Additionally, these are viewed in the context of IMS to see whether they could be used to enhance the security properties of such service architecture.

While the host identities have a considerable effect on solving the dual nature of IP addresses by providing an additional entity identification layer, the profound security value comes from the possibility of having an identity, for which a proof can be provided. In essence, we are able to name the end entities in a secure fashion and bind assured statements to these entities. Thus, accountability of the actions becomes easier as it is possible to provide assured evidence that an entity was involved in an action.

Host identities come with an accompanying handshake protocol for identity and key establishment. In this thesis, several cases have been presented on how that procedure can be overloaded to enable additional functionalities. The foremost application is in the network attachment, where it has been used as a network attachment protocol. Basically, configuration of end entities can be protected using the same procedure, but some additional signalling information could be transferred to enhance the performance of the attachment procedure. For instance, signalling could be delegated to the network elements closer to the core in order to avoid unnecessary use of the wireless interface. Such delegation, in addition to other authorisation statements, can be employed in a natural way due to the existence of secure identifiers. While this work has mainly concentrated on HIP in its solution space, the possibility of employing similar principles to devise a common attachment procedure with consistent security mechanisms for the future networks exists. Such a holistic approach could provide a cross-layered solution, which would make it possible to avoid having a different security infrastructure on every layer.

The other major point of this thesis considered the accounting aspects of the service usage. This does not just entail services on the application layer, but considers also network connectivity as a service. In any case, the aim of this thesis was to consider a solution for ensuring that both parties of the service provisioning have control over what they are committing themselves to. While we have not considered monetary issues per se, eventually the service provisioning transfers the accounting figures into money, that is, compensation for the resource usage. With assured identities one is able to bind the accounting to the correct entities in a reliable way, so that one is not able to deny its

participation at a later stage. The possibility to attach statements to the identities also gives the possibility to introduce better risk management mechanisms. In other words, instead of having an all or nothing approach, the service provisioning can be viewed from a granular perspective and the evidence provided in a piece-meal fashion. Thus, both the consumer and the provider of the service can track whether the other party is acting as expected. If not, the interaction can be terminated and no further commitment to the resource compensation or provisioning is needed. However, one should note that this approach only works for services, which can be serialised in a natural way, which is, it is meaningful to provide them in a granular fashion and ascertain that the service is proceeding.

One strong background theme in the course of this research has been the future networking landscape with its ubiquitous computing and ambient intelligence ideas. While that promotes seamlessness, it also requires more interaction with previously unknown entities. This can be especially troublesome for incumbent operators, who have become used to the idea of static and well-controlled interactions. In this light the previous topics can be used to provide technical assurance mechanisms for this dilemma, even though from economical and political perspectives it can still have other difficult problems. Assured identification can be used to account for the actions of the subscribers and additional, traceable statements can be used to control the level of allowed actions. Also, granular approaches can be used to track the usage when the signalling is routed via the home operator as well. This can have the incentive of allowing the optimisation of the actual traffic. This thesis has investigated these points within the context of IMS and shown how the current architecture could be used to provide such enhanced assurance properties. The cross-layered approach also shows how the availability of the secure identifiers can be used to bind different parts of the architecture for better assurance of the whole service provisioning. In other words, the actions taken during the initial attachment can be used to provide assurance to the agreement processes between other entities, for example, the roaming agreement negotiation between the visited and home operators. Additionally, involving the operator provides freshness qualities, which can be used in aiding revocation concerns (even though one might argue that usually the cause of revocation is not noticed until the charges are due).

One could naturally question the feasibility of applying the host identities in such various ways, especially knowing that originally they have been intended to be network stack specific constructs. Thus, they are more tied to the network element rather than to an individual. However, the very existence of host identity (assuming that HIP eventually “makes a breakthrough”) can be used as a basis for creating a crypto-identity based security ecosystem. It is still natural to use such identities to account for the actions related to the network level, but also additional higher layer identities can be bound to them. They could be cryptographical in nature as well or more along the lines of IMS application level identities. The existing authentication mechanisms, albeit with modifications, can then be applied to register the binding between the different identities, pro-

vided that the registration procedure is termed secure enough, that is, preferably not based solely on typical passwords. However, one should remember that in subsequent interaction the authentication and authorisation decisions should be based on the crypto-identities and the display of the proof of possession. One additional point to remember is the duration of the binding, that is, for the sake of privacy and risk management, that should not be overly long.

The ideas in this thesis can be said to be visionary ones as there is really no concrete proof that the presented assumptions will become reality. In addition, the presented architectures have been largely left unimplemented so far, even though similar ideas have been implemented in other research projects (due to common “project ancestry”). That partly shows the reason as well: this work has been mainly an individual effort with no backing of a well resourced project. Thus, the ideas should be further refined in future collaborative efforts and concrete implementations provided to show the feasibility of the approach. One should note that even though HIP might not become a prevalent mechanism in the Internet, one could still expect that eventually the entities will have a strong identity, which will provide the basis of accountability. Whether such identity (or identities) is provided in software, hardware, or in a separate authentication device in the fashion of a SIM card, is still left to be seen.

BIBLIOGRAPHY

[23.003] 3GPP, “Numbering, addressing and identification”, 3GPP Technical Specification TS 23.003 V10.0.0, Dec 2010.

[23.060] 3GPP, “General Packet Radio Service (GPRS); Service description”, 3GPP Technical Specification TS 23.060 V10.3.0, Mar 2011.

[23.228] 3GPP, “IP Multimedia Subsystem (IMS)”, 3GPP Technical Specification TS 23.228 V11.0.0, Mar 2011.

[23.234] 3GPP, “3GPP system to Wireless Local Area Network (WLAN) interworking; System description”, 3GPP Technical Specification TS 23.234 V10.0.0, Mar 2011.

[23.401] 3GPP, “General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access”, 3GPP Technical Specification TS 23.401 V10.3.0, Mar 2011.

[24.229] 3GPP, “IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)”, 3GPP Technical Specification TS 24.229 V10.2.0, Dec 2010.

[32.260] 3GPP, “IP Multimedia Subsystem (IMS) charging”, 3GPP Technical Specification TS 32.260 V10.3.0, Mar 2011.

[33.102] 3GPP, “3G Security; Security architecture”, 3GPP Technical Specification TS 33.102, V10.0.0, Dec 2010.

[33.107] 3GPP, “Lawful interception architecture and functions”, 3GPP Technical Specification TS 33.107, V10.3.0, Mar 2011.

[33.203] 3GPP, “Access security for IP-based services”, 3GPP Technical Specification TS 33.203 V11.0, Dec 2010.

[33.210] 3GPP, “Network Domain Security (NDS); IP network layer security”, 3GPP Technical Specification TS 33.120, V11.0.0, Dec 2010.

[33.220] 3GPP, “Generic Authentication Architecture (GAA);Generic Bootstrapping Architecture (GBA)”, 3GPP Technical Specification TS 33.220, V10.0.0, Oct 2010.

[33.221] 3GPP, “Generic Authentication Architecture (GAA); Support for subscriber certificates”, 3GPP Technical Specification TS 33.221 V10.0.0, Mar 2011.

[33.310] 3GPP, “Authentication Framework (AF)”, 3GPP Technical Specification TS 33.310 V10.2.0, Dec 2010.

[802.11] IEEE Computer Society, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE Std 802.11™-2007, Jun 2007.

[802.11w] IEEE Computer Society, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications; Amendment 4: Protected Management Frames”, IEEE Std 802.11w™-2009, Sep 2009.

[802.1AR] IEEE Computer Society, “Secure Device Identity”, IEEE Std 802.1AR™-2009, Dec 2009.

[802.1X] IEEE Computer Society, “Port-Based Network Access Control, IEEE Std 802.1Z™-2010, Feb 2010.

[Abd00] Abdul-Rahman A., Hailes S., “Supporting Trust in Virtual Communities”, Proceedings of the 33rd Hawaii International Conference on System Sciences, Jan 2000.

[Abo00] Aboba B., Arkko J., Harrington D., “Introduction to Accounting Management”, IETF RFC 2975, Oct 2000.

[Abo03] Aboba B., Calhoun P., “RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)”, IETF RFC 3579, Sep 2003.

[Abo04] Aboba B., Blunk L., Vollbrech J., Carlson J., Levkowetz H., “Extensible Authentication Protocol (EAP)”, IETF RFC 3748, Jun 2004.

[Abo05] Aboba B., Beadles M., Arkko J., Eronen P., “The Network Access Identifier”, IETF RFC 4282, Dec 2005.

[Ada04] Adams C., Just M., “PKI: Ten Years Later”, Proceedings of 3rd Annual PKI R&D Workshop, Apr 2004.

[Ale97] Alexander S., Droms R., “DHCP Options and BOOTP Vendor Extensions”, IETF RFC 2132, Mar 1997.

[AlR03] Al-Riyami S., Paterson K., “Certificateless Public Key Cryptography”, Advances in Cryptology - ASIACRYPT 2003, Lecture Notes in Computer Science, Vol 2894/2003, pp. 452-473, 2003.

[Amo94] Amoroso E., “Fundamentals of computer security technology”, Prentice-Hall. 1994.

[An07] An G., Kim K., Jang J., Jeon Y., “Analysis of SEND Protocol through Implementation and Simulation”, Proceedings of International Conference on Convergence Information Technology, Nov 2007.

[And94] Anderson R., “Liability and Computer Security: Nine principles”, Proceedings of the Third European Symposium on Research in Computer Security, Nov 1994.

[And08] Andersen D. G. et al., “Accountable Internet Protocol (AIP)”, Proceedings of ACM SIGCOMM, Aug 2008.

[Are05] Arends R., Austein R., Larson M., Massey D., Rose S., “DNS Security Introduction and Requirements”, IETF RFC 4033, Mar 2005.

[Ark04a] Arkko J., Carrara E., Lindholm F.,Naslund M., Norrman K., “MIKEY: Multimedia Internet KEYing”, IETF RFC 3830, Aug 2004.

[Ark04b] Arkko J., Nikander P., Eronen P., Torvinen V., ”Secure and Efficient Network Access”, DIMACS Workshop on Mobile and Wireless Security, Nov 2004.

[Ark05] Arkko J. (Ed.), Kempf J., Zill B., Nikander P., “SEcure Neighbor Discovery (SEND)”, IETF RFC 3971, Mar 2005.

[Ark06] Arkko J., Eronen P., Tschofenig H., Heikkinen S., Prasad A. ” Quick NAP - Secure and Efficient Network Access Protocol”, Proceedings of the 6th International Workshop on Applications and Services in Wireless Networks, May 2006.

[Ark07] Arkko J., Vogt C., Haddad W., “Enhanced Route Optimization for Mobile IPv6”, IETF RFC 4866, May 2007.

[Aso98] Asokan N., “Fairness in Electronic Commerce”, PhD thesis, University of Waterloo, Canada, 1998.

[Aso00] Asokan N., Shoup V., Waidner M., “Optimistic Fair Exchange of Digital Signatures”, IEEE Journal on Selected Areas in Communications, Vol 18, No 4, pp. 593-610, Apr 2000.

[Aud09] Audet F., The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)”, IETF RFC 5630, Oct 2009.

[Aur05a] Aura T., “Cryptographically Generated Addresses (CGA)”, IETF RFC 3972, Mar 2005.

[Aur05b] Aura T., Nagarajan A, Gurtov A., “Analysis of HIP Base Exchange Protocol”, Proceedings of 10th Australasian Conference on Information Security and Privacy (ACISP 2005), Jul 2005.

[Bag09] Bagnulo M., “Hash-Based Addresses (HBA)”, IETF RFC 5535, Jun 2009.

[Bel76] Bell D. E., La Padula L. J., “Secure Computer System: Unified Exposition and Multics Interpretation”, United States Air Force Project 522B Report ESD-TR-75-306, Mar 1976.

[Bis02] Bishop M., “Computer Security: art and science”, Addison-Wesley, 2002

[Bla02] Blaze M., Ioannidis J., Keromytis A., “Offline Micropayments without Trusted Hardware”, Proceedings of the 5th International Conference on Financial Cryptography, Feb 2002.

[Bla03] Blaze M. at al., “TAPI: Transactions for Access Public Infrastructure. Proceedings of Personal Wireless Communications”, Proceedings of the 8th IFIP Personal Wireless Communications Conference, Sep 2003.

[Blo70] Bloom B. H., “Space/time trade-offs in hash coding with allowable errors”, Communications of the ACM, Vol 13, Iss 7, pp. 422-426, Jul 1970.

[Blu83] Blum M., “How to Exchange (Secret) Keys”, ACM Transactions on Computer Systems, Vol 1, No 2, pp. 175-193, May 1983.

[Bon01] Boneh D., Franklin M., “Identity-Based Encryption from the Well Pairing”, Advances in Cryptology - CRYPTO 2001, Lecture Notes in Computer Science, Vol 2139/2001, pp. 213-229, 2001.

[Bor01] Borisov N., Goldberg I., Wagner D., “Intercepting mobile communications: the insecurity of 802.11”, Proceedings of the 7th annual international conference on Mobile computing and networking, Jul 2001.

[Bra10] Bratus S., Lembree A., Shubina A., “Software on the witness stand: what should it take for us to trust it?”, Proceedings of the 3rd international conference on Trust and trustworthy computing, Jun 2010.

[Bre02] Brezak J., “Utilizing the Windows 2000 Authorization Data in Kerberos Tickets for Access Control to Resources”, MSDN Library, Feb 2002, available at http://msdn.microsoft.com/en-us/library/aa302203.aspx (checked 10/2011).

[Bru03] Bruschi D, Ornaghi A., Rosti E., “S-ARP: a Secure Address Resolution Protocol”, Proceedings of the 19th Annual Computer Security Applications Conference, Dec 2003.

[Cal03] Calhoun P., Loughney J., Guttman E., Zorn G., Arkko J., “Diameter Base Protocol”, IETF RFC 3588, Sep 2003.

[Cam02] Camarillo G., Marshal W., Rosenberg J., “Integration of Resource Management and Session Initiation Protocol (SIP)”, IETF RFC 3312, Oct 2002.

[Cam06] Camarillo G., Garcia-Martin M.A., “The 3G IP Multimedia System (IMS), Second Edition”, Wiley, 2006.

[Cam11] Camarillo G., Melen J., “Host Identity Protocol (HIP) Immediate Carriage and Conveyance of Upper-Layer Protocol Signaling (HICCUPS)”, IETF RFC 6078, Jan 2011.

[Can05] Candolin C., Lundberg J.,Kari H., “Packet level authentication in military networks”, Proceedings of the 6th Australian Information Warfare & IT Security Conference, Nov 2005.

[Cas04] Castellucia C., Montenegro G., Laganier J., Neumann C., “Hindering Eavesdropping via IPv6 Opportunistic Encryption”, Proceedings of the European Symposium on Research in Computer Security, Sep 2004.

[CCA09] The Common Criteria Recognition Arrangement, “Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components”, CCMB-2009-07-03, Jul 2009.

[Che05] Cheshire S., Aboba B., Guttman E., “Dynamic Configuration of IPv4 Link-Local Addresses”, IETF RFC 3927, May 2005.

[Che08] Cheshire S., “IPv4 Address Conflict Detection”, IETF RFC 5227, Jul 2008.

[Con06] Conta A., Deering S., Gupta M. (Ed.), “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification”, IETF RFC 4443, Mar 2006.

[Coo08] Cooper D. et al., “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, IETF RFC 5280, May 2008.

[Dee91] Deering S. (Ed.), “ICMP Router Discovery Messages”, IETF RFC 1256, Sep 1991.

[Del09] Delamaire L., Abdou H., Pointon J., “Credit card fraud and detection techniques: a review”, Banks and Bank Systems, Vol 4, Iss 2, pp. 57-68, 2009.

[Die08] Dierks T, Rescorla E., “The Transport Layer Security (TLS) Protocol Version 1.2”, IETF RFC 5246, Aug 2008.

[Dro97] Droms R., “Dynamic Host Configuration Protocol”, IETF RFC 2131, Mar 1997.

[Dro01] Droms. R. (Ed.), Arbaugh W. (Ed.), “Authentication for DHCP Messages”, IETF RFC 3118, Jun 2001.

[Dro03] Droms R. (Ed.), “Dynamic Host Configuration Protocol for IPv6 (DHCPv6)”, IETF RFC 3315, Jul 2003.

[Dro04] Droms R., “Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6”, IETF RFC 3736, April 2004.

[Dub04] Dübendorfer T., Wagner A., Plattner B., “An Economic Damage Model for Large-Scale Internet Attacks”, Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Jun 2004.

[Dur00] Durham D. (Ed.), “The COPS (Common Open Policy Service) Protocol”, IETF RFC 2748, Jan 2000.

[EC0258] European Parliament and the Council, “Directive on privacy and electronic communications (2002/58/EC)”, Directive 2002/58/EC of the European Parliament and of the Council, Jul 2002.

[EC96] The Council of the European Union, “Council resolution of 17 January 1995 on the lawful interception of telecommunications (96/C 329/01)”, Official Journal of the European Communities, Nov 1996.

[Ell99] Ellison C. (Ed.), “Simple Public Key Certificate”, Internet Draft draft-ietf-spki-cert-structure-06 (expired), Jul 1999.

[Ell00] Ellison C., Schneier B., “Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure”, Computer Security Journal, Vol 16, No 1, pp. 1-7, 2000.

[ElS08] El Sawda S., El Sawda R., Urien P., Hajjeh I., "Non Repudiation for SIP Protocol; SIP Sign", Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications, Apr 2008.

[Ero04] Eronen P., Arkko J., ”Role of authorization in wireless network security”, DIMACS Workshop on Mobile and Wireless Security, Nov 2004.

[Far10] Farrell S., Housley R., Turner S., “An Internet Attribute Certificate Profile for Authorization”, IETF RFC 5755, Jan 2010.

[Fie99] Fielding R. et al., “Hypertext Transfer Protocol -- HTTP/1.1”, IETF RFC 2616, Jun 1999.

[Flu01] Fluhrer S., Mantin I., Shamir A., “Weaknesses in the Key Scheduling Algorithm of RC4”, Proceedings of 8th Annual International Workshop on Selected Areas in Cryptography, 2001.

[For08] Forsberg D., Ohba Y. (Ed.), Patil B., Tschofenig H., Yegin A.,”Protocol for Carrying Authentication for Network Access”, IETF RFC 5191, May 2008.

[Gai07] Gaitan O.S., Martins P., Tohme S., Demerjan J., “SIP Embedded Attribute Certificates For Service Mobility in Heterogeneous Multi-Operator Wireless Networks”, Proceedings of the 66th IEEE Vehicular Technology Conference, Sep 2007.

[Gra78] Gray J., “Notes on Data Base Operating Systems”, Operating Systems, Advanced Course, Lecture Notes on Computer Science, Vol 60, pp. 393-481, 1978.

[Gra00] Grandison T., Sloman M., ”A Survey of Trust in Internet Applications”, IEEE Communications Surveys and Tutorials, Vol 3, Iss 4, pp. 2-16, Sep 2000.

[Gur03] Gürgens S., Rudolp C., “Security Analysis of (Un-) Fair Non-repudiation Protocols”, Format Aspects of Security, Lecture Notes in Computer Science, Vol 2629, pp. 97-114, 2003.

[Gur05] Gürgens S., Rudolp C., Vogt H., “On the Security of Fair Non-repudiation Protocols”, International Journal of Information Security, Vol 4, Iss 4, pp. 193-207, Oct 2005.

[Gur08] Gurtov A., “Host Identity Protocol (HIP): Towards the Secure Mobile Internet”, Wiley, 2008.

[Gus03] Gustafsson E., Jonsson A., “Always best connected”, IEEE Wireless Communications, Vol 10, Iss 1, pp. 49-55, Feb 2003.

[Gut04] Gutmann P., “Why isn’t the Internet secure yet, dammit”, In AusCERT Asia Pacific Information Technology Security Conference 2004, May 2004.

[Gär99] Gärtner F., Pagnia H., Vogt H., “Approaching a Formal Definiotion of Fairness in Electronic Commerce”, Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems, Oct 1999.

[Hal11] Hallam-Baker P., “The Recent RA Compromise”, Comodo Data Security blog entry, Mar 2011, available from http://blogs.comodo.com/it-security/data-security/the-recent-ra-compromise/ (checked 10/2011).

[Han06] Handley M., Jacobson V., Perkins C., SDP: Session Description Protocol”, IETF RFC 4566, Jul 2006.

[Har02] Harrington D., Presuhn R., Wijnen B., “An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks”, IETF 3411, Dec 2002.

[Hau08] Hautakorpi J., Salinas A., Harjula E., Ylianttila M., ”Interconnecting P2PSIP and IMS”, Proceedings of the Second International Conference on Next Generation Mobile Applications, Services, and Technologies, Sep 2008.

[Hee07] Heer T., “LHIP Lightweight Authentication Extension for HIP”, Internet Draft draft-heer-hip-lhip-00 (expired), Feb 2007.

[Hee09] Heer T., Hummen R., Komu M., Kötz S., Wehrle K., ”End-host Authentication and Authorization for Middleboxes based on a Cryptographic Namespace”, Proceedings of the IEEE International Conference on Communications, Jun 2009.

[Hei04] Heikkinen S., Tschofenig H., Gelbord B., ”Network Attachment and Address configuration using HIP”, Position paper in Workshop on HIP and Related Architectures, Nov 2004.

[Hei05] Heikkinen S., Priestley M., Arkko J., Eronen P.,Tschofenig H., ”Securing Network Attachment and Compensation”, Proceedings of Wireless World Research Forum Meeting (WWRF#15), Nov 2005.

[Hei06] Heikkinen S., “Social engineering in the world of emerging communication technologies”, Proceedings of Wireless World Research Forum Meeting (WWRF#17), Nov 2006.

[Hei09] Heikkinen S., “Providing Identity Assured User Generated Services Using IMS”, Proceedings of the 2nd International Workshop on Mobile and Wireless Networks Security, May 2009.

[Her95] Herda S., “Non-repudiation: Constituting evidence and proof in digital cooperation”, Computer Standards & Interfaces, Vol 17, Iss 1, pp. 69-79, Jan 1995.

[Hib06] Hibbs R., Smith C., Volz B., Zohar M., “Dynamic Host Configuration Protocol for IPv4 (DHCPv4) Threat Analysis”, IETF Internet-Draft draft-ietf-dhc-v4-threat-analysis-03.txt (expired), Jun 2006.

[Hin06] Hinden R., Deering S., “IP Version 6 Addressing Architecture”, IETF RFC 4291, Feb 2006.

[Hof02] Hoffman P. (Ed.), “Features of Proposed Successors to IKE”, Internet Draft draft-ietf-ipsec-soi-features-01 (expired), May 2002.

[Im08] Im T. R., Lee H., Cho K. T., Lee D. H., “Secure Mutual Authentication and Fair Billing for Roaming Service in Wireless Mobile Networks”, Proceedings of the Third International Conference on Convergence and Hybrid Information Technology, Nov 2008.

[Jay08] Jayaraman P., Lopez R., Ohba Y., Parthasarathy M., Yegin A., “Protocol for Carrying Authentication for Network Access (PANA) Framework”, IETF RFC 5193, May 2008.

[Jen02] Jennings C., Peterson J., Watson M., “Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks”, IETF RFC 3325, Nov 2002.

[Jen11] Jennings C., Fischl J., “Certificate Management Service for the Session Initiation Protocol (SIP)”, IETF RFC 6072, Feb 2011.

[Jer08] Jerschow Y., Lochert C., Scheuermann B., Mauve M., “CLL: A Cryptographic Link Layer for Local Area Networks”, Proceedings of the 6th international conference on Security and Cryptography for Networks, 2008.

[Jia11] Jiang S., Xia S., ”Configuring Cryptographically Generated Addresses (CGA) using DHCPv6”, Internet Draft draft-ietf-dhc-cga-config-dhcpv6-00 (work in progress), Apr 2011.

[Jok08] Jokela P., Moskowitz R., Nikander P., ”Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)”, IETF RFC 5202, Apr 2008.

[Kai96] Kailar R., “Accountability in Electronic Commerce Protocols”, IEEE Transactions on Software Engineering, Vol 22, No 5, pp. 313-328, May 1996.

[Kap07] Kappler C., Pöyhönen P., Johnsson M., Schmid S., ”Dynamic Network Composition for Beyond 3G Networks: A 3GPP Viewpoint”, IEEE Network, Vol 21, Iss 1, pp. 47-52, Jan 2007.

[Kau10] Kaufman C., Hoffman P., Nir Y., Eronen P., ”Internet Key Exchange Protocol Version 2 (IKEv2)”, IETF RFC 5996, Sep 2010.

[Kem06] Kempf J., Wood J., Ramzan Z, Gentry C., “IP Address Authorization for Secure Address Proxying Using Multi-key CGAs and Ring Signatures”, Advances in Information and Computer Security, Lecture Notes in Computer Science, Volume 4266, pp. 196-211, 2006.

[Kjä09] Kjällman, J., ”Attachment to a Native Publish/Subscribe Network”, Procedings of International Workshop on the Network of the Future, Jun 2009.

[Kle08] Klensin J., “Simple Mail Transfer Protocol”, IETF RFC 5321, Oct 2008.

[Kre02] Kremer S., Markowitch O., Zhou J., “An Intensive survey of fair non-repudiation protocols”, Computer Communications Vol 25, pp. 1606-1621, Jan 2002.

[Kom05] Komu M., Tarkoma S., Kangasharju J., Gurtov A., ”Applying a Cryptographic Namespace to Applications”, Proceedings of the 1st ACM workshop on Dynamic interconnection of networks, Sep 2005.

[Kom10] Komu M., Henderson T., Tschofenig H., Melen J., Keränen A. (Ed.), ” Basic Host Identity Protocol (HIP) Extensions for Traversal of Network Address Translators”, IETF RFC 5770, Apr 2010.

[Kop05] Koponen T., Gurtov A., Nikander P., ”Application mobility with Host Identity Protocol”, Proceedings of NDSS Wireless and Mobile Security Workshop, Feb 2005.

[Kor07] Korhonen J., Mäkela A., Rinta-aho T., “HIP Based Network Access Protocol in Operator Network Deployments”, Proceedings of First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management, Oct 2007.

[Kut07] Kutvonen L., Metso J., Ruohomaa S., ”From trading to eCommunity management: Responding to social and contractual challenges”, Information Systems Frontiers, Vol 9, Iss 2, pp. 181-194, 2007.

[Lag05] Laganier J., Vicat-Blanc Primet P., “HIPernet: A Decentralized Security Infrastructure for Large Scale Grid Environments”, Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing, Nov 2005.

[Lag07] Laganier J., Montenegro G., Kukec A., “Using IKE with IPv6 Cryptographically Generated Addresses”, Internet Draft draft-laganier-ike-ipv6-cga-02 (expired), Jul 2007.

[Lag08] Laganier J., Eggert L., “Host Identity Protocol (HIP) Rendezvous Extension”, IETF RFC 5204, Apr 2008.

[Lag10] Lagutin D., “Securing the Internet with digital signatures”, PhD Thesis, Aalto University, School of Science and Technology, Espoo, Nov 2010.

[Lam81] Lamport L., “Password Authentication with Insecure Communication”, Communications of the ACM, Vol 24, No 11, pp. 770-772, Nov 1981.

[LAPS03] Varney C. (Ed.), “Privacy and Security Best Practises”, Liberty Alliance white paper, Nov 2003.

[Lem06] Lemon T., Sommerfield B., “Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)”, IETF RFC 4361, Feb 2006.

[Li09] Li S., Wang G., Zhou J., Chen K., “Fair and Secure Mobile Billing Systems”, Wireless Personal Communications, Vol 51, No 1, pp. 81-93, Oct 2009.

[Loo07] Lootah W., Enck W., McDaniel P., “TARP: Ticket-based address resolution protocol”, Computer Networks, Vol 51, Iss 15, pp. 4322-4337, Oct 2007.

[Lou00] Louridas P., “Some guidelines for non-repudiation protocols”, ACM SIGCOMM Computer Communication Review, Vol 30, Iss 5, pp. 29-38, Oct 2000.

[Low10] Lowther D., “GSMA Leads Mobile Industry Towards a Single, Global Solution for Voice over LTE”, GSMA press release 15.2.2010, Feb 2010, available at http://www.gsm.org/newsroom/press-releases/2010/4634.htm (checked 10/2011).

[Mar03] Marshall W. (Ed.), “Private Session Initiation Protocol (SIP) Extensions for Media Authorization”, IETF RFC 3313, Jan 2003.

[Mar07] Markendahl J., Johnsson M., “Ambient networking and related business concepts as support for regulatory initiatives and competition”, Netnomics, Vol 8, Iss 1-2, pp. 105-121, Nov 2007.

[Mat07] Matuszewski M., Garcia-Martin M.A., “A Distributed IP Multimedia Subsystem (IMS)”, Proceedings of IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks, Jun 2007.

[May91] Mayfield T., Roskos J.E., Welke S.R., Boone J.M., “Integrity in automated information systems”, National Computer Security Center (NCSC) Technical Report 79-91, Sep 1991.

[Mit01] Mitton D. et al., ”Authentication, Authorization, and Accounting: Protocol Evaluation”, IETF RFC 3127, Jun 2001.

[Mon02] Montenegro G., Castellucia C., “Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses”, In Proceedings of the 9th Annual Network and Distributed System Security Symposium, Feb 2002.

[Mos03] Moskowitz R., “Weakness in Passphrase Choice in WPA Interface”, Wi-Fi Net News online article, Nov 2003, available at http://wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html (checked 10/2011)

[Mos06] Moskowitz R., Nikander P., “Host Identity Protocol (HIP) Architecture”, IETF RFC 4423, May 2006.

[Mos08] Moskowitz R., Nikander P., Jokela P. (Ed.), Henderson T., “Host Identity Protocol”, IETF RFC 5201, Apr 2008.

[Mos11] Moskowitz R., “HIP Diet Exchange (DEX)”, Internet Draft draft-moskowitz-hip-rg-dex-05 (work in progress), Mar 2011.

[Mul86] Mullender S. J., Tanenbaum A. S., “The Design of a Capability-Based Distributed Operating System”, The Computer Journal, Vol 29, No 4, pp. 289-299, 1986.

[Nar07a] Narten T., Nordmark E., Simpson W., Soliman H., “Neighbor Discovery for IP version 6 (IPv6)”, IETF RFC 4861, Sep 2007.

[Nar07b] Narten T., Draves R., Krishnan S., “Privacy Extensions for Stateless Address Autoconfiguration in IPv6”, IETF RFC 4941, Sep 2007.

[Neu05] Neuman C., Yu T., Hartman S., Raeburn K., ”The Kerberos Network Authentication Service (V5)”, IETF RFC 4120, Jul 2005.

[Nie02] Niemi V., Arkko J., Torvinen V., ”Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)”, IETF RFC 3310, Sep 2002.

[Nie05] Niebert N., “Ambient networks: a framework for mobile network cooperation”, Proceedings of the 1st ACM workshop on Dynamic interconnection of networks, Sep 2005.

[Nie07] Niebert N., Schieder, Zander J., Hancock R., (Eds.), “Ambient Networks – Co-operative Mobile Networking for the Wireless World”, Wiley, 2007.

[Nik01] Nikander P., “An Address Ownership Problem in IPv6”, Internet Draft draft-nikander-ipng-address-ownership-00 (expired), Feb 2001.

[Nik04a] Nikander P. (Ed.), Kempf J., Nordmark E., “IPv6 Neighbor Discovery (ND) Trust Models and Threats”, IETF RFC 3756, May 2004.

[Nik04b] Nikander P., Arkko J., Ohlman B., “Host Identity Indirection Infrastructure (Hi3)”, Proceedings of Swedish National Computer Networking Workshop, Nov 2004.

[Nik07] Nikander P., Laganier J., Dupont F., “An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID)”, IETF RFC 4843, Apr 2007.

[Nik08] Nikander P., Henderson T. (Ed.), Vogt C., Arkko J., “End-Host Mobility and Multihoming with the Host Identity Protocol”, IETF RFC 5206, Apr 2008.

[NIST01] Stoneburner G. (Ed.), “Underlying Technical Models for Information Technology Security”, NIST Special Publication 800-33, Dec 2001.

[OAS05] OASIS, “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0”, OASIS Standard, Mar 2005.

[OSh01] O’Shea G., Roe M., “Child-proof Authentication for MIPv6 (CAM)”, ACM SIGCOMM Computer Communication Review, Vol 31, Iss 2, pp. 4-8, Apr 2001.

[Owi82] Owicki S., Lamport L., “Proving Liveness Properties of Concurrent Programs”, ACM Transactions on Programming Languages and Systems, Vol 4, No 3, pp. 455-495, Jul 1982.

[Pap03] Papazoglou M.P., Georgakopoulos D., “Service Oriented Computing”, Communications of ACM, Vol 46, No 10, pp. 25-28, Oct 2003.

[Par05] Parthasarathy M., “PANA Enabling IPsec based Access Control”, Internet Draft draft-ietf-pana-ipsec-07 (expired), Jul 2005.

[Pea02] Pearlman L., Welch V., Foster I., Kesselman C., Tuecke S., “A Community Authorization Service for Group Collaboration”, Proceedings of the third International workshop on Policies for Distributed Systems and Networks, Jun 2002.

[Pet06] Peterson J., Jennigs C., “Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)”, IETF RFC 4474, Aug 2006.

[Plu82] Plummer D., “An Ethernet Address Resolution Protocol”, IETF RFC 826, Nov 1982.

[Pos81] Postel J., “Internet Control Message Protocol”, IETF RFC 792, Sep 1981.

[Pra05] Pras A., van de Meent R., Mandjes M., ”QoS in Hybrid Networks - An Operator’s Perspective”, Proceedings of the 13th IEEE International Workshop on Quality of Service, Jun 2005.

[Pre07] Prevelakis V., Spinellis D., “The Athens Affair”, IEEE Spectrum, Vol 44, Iss 7, pp. 26-33, Jul 2007.

[Rag04] Raghunathan V., Pering T., Want R., Nguyen A., Jensen P., “Experience with a low power wireless mobile computing platform”, Proceedings of the 2004 International Symposium on Low Power Electronics and Design, Aug 2004.

[Ram10] Ramsdel B., Turner S., “Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification”, IETF RFC 5751, Jan 2010.

[Rig00] Rigney C., Willens S., Rubens A., Simpson W., “Remote Authentication Dial in User Service (RADIUS)”, IETF RFC 2865, Jun 2000.

[Rin06a] Ring J., Choo K., Foo E., Looi M., “A New authentication Mechanism and Key Agreement Protocol for SIP Using Identity-based Cryptography”, Proceedings of AusCERT Asia Pacific Information Technology Security Conference, May 2006.

[Rin06b] Rinta-aho T. et al., ”Ambient Network Attachment”, Proceedings of 16th IST Mobile & Wireless Communications Summit, Jul 2006.

[Riv96] Rivest R., Shamir A., “PayWord and MicroMint: Two Simple Micropayment Schemes”, Proceedings of International Workshop on Security Protocols, Apr 1996.

[Riv01] Rivest R., Shamir A., Tauman Y., “How to Leak a Secret: Theory and Applications of Ring Signatures”, Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, Jun 2001.

[Ros02a] Rosenberg J. et al., ”SIP: Session Initiation Protocol”, IETF RFC 3261, Jun 2002.

[Ros02b] Rosenberg J., Schulzrinne H., “Reliability of Provisional Responses in the Session Initiation Protocol (SIP)”, IETF RFC 3262, Jun 2002.

[Rui06] Ruiz-Martinez A., Marin-Lopez C.I., Bano-Lopez L., Gomez Skarmeta A.F., ”A new fair non-repudiation protocol for secure negotiation and contract signing”, Proceedings of the 2006 International Conference on Privacy, Security and Trust, Nov 2006.

[Ruo10] Ruohonen S., Kutvonen L., ”Trust and Distrust in Adaptive Inter-enterprise Collaboration Management”, Journal of Theoretical and Applied Electronic Commerce Research, Vol 5, Iss 2, pp. 118-136, Aug 2010.

[Räi05] Räisänen V., Kellerer W., Hölttä P., Karasti O., Heikkinen S., ”Service Management Evolution”, Proceedings of the 14th IST Mobile & Wireless Communications Summit, Jun 2005.

[Sch02] Schulzrinne H., “Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers”, IETF RFC 3361, Aug 2002.

[Sch03] Schulzrinne H., “Dynamic Host Configuration Protocol (DHCPv6) Option for Session Initiation Protocol (SIP) Servers”, IETF RFC 3319, Jul 2003.

[Sch09] Schridde C., Smith M., “Freisleben B., “TrueIP: Prevention of IP Spoofing Attacks Using Identity-Based Cryptography”, Proceedings of the 2nd international conference on Security of information and networks, Oct 2009.

[Sei04] Seigneur J., Farrell S., Jensen C., Gray E., Chen Y., “End-to-end Trust Starts with Recognition”, Security in Pervasive Computing, Lecture Notes in Computer Science, Vol 2802, pp. 130-142, 2004.

[Sel05] Selander G. (Ed.), “Ambient Network Intermediate Security Architecture”, Ambient Network WP7 deliverable IST–2002-507134-AN/ WP7/D01, Jan 2005.

[Sha85] Shamir A., “Identity-based cryptosystems and signature schemes”, Advances in cryptology: Proceedings of CRYPTO 84, 1985.

[Shi07] Shirey R., “Internet Security Glossary, Version 2”, IETF RFC 4949, Aug 2007.

[Sin05] Singh K., Schulzrinne H., “Peer-to-Peer Internet Telephony using SIP”, Proceedings of the International workshop on Network and operating systems support for digital audio and video, Jun 2005.

[Sim08] Simon D, Aboba B, Hurst R., “The EAP-TLS Authentication Protocol”, IETF RFC 5216, Mar 2008.

[Sta98] Stallings W., “Cryptography and network security: principles and practice,”, Second edition, Prentice Hall, 1998.

[Sär08] Särelä M., Rinta-aho T., Tarkoma S., ”RTFM: Publish/Subscribe Internetworking Architecture”, Proceedings of ICT-MobileSummit 2008, Jun 2008.

[Tar07] Tarkoma S. et al., ” Spice: A Service Platform for Future Mobile IMS Services”, Proceedings of International Symposium on a World of Wireless, Mobile and Multimedia Networks, Jun 2007.

[Tch06] Tschofenig H. et al., “Using SAML to protect the session initiation protocol (SIP)”, IEEE Network, Vol 20, Iss 5, pp. 14-17, Sep 2006.

[Tew03] Tewari H., O’Mahon D., “Multiparty Micropayments for Ad Hoc Networks”, Proceedings of the IEEE Wireless Communications and Networking Conference, Mar 2003.

[Tew09] Tews E., Beck M., “Practical attacks against WEP and WPA”, Proceedings of the second ACM conference on Wireless network security, Mar 2009.

[Tho07] Thomson S., Narten T., Jinmei T., “IPv6 Stateless Address Autoconfiguration”, IETF RFC 4862, Sep 2007.

[Tou08] Touch J., Black D., Wang Y., “Problem and Applicability Statement for Better-Than-Nothing Security (BTNS)”, IETF RFC 5387, Nov 2008.

[Tse04] Tseng Y., Yang C., Su J., ”Authentication and Billing Protocols for the Integration of WLAN and 3G Networks”, Wireless Personal Communications, Vol 29, Iss 3-4, pp. 351-366, Jun 2004.

[Tue04] Tuecke S., Welch V., Engert D., Pearlman L., Thomson M., “Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile”, IETF RFC 3820, Jun 2004.

[Wan08] Wang F., Zhang Y., “A New Provably Secure Authentication and Key Agreement Mechanism for SIP Using Certificateless Public-key Cryptography”, Computer Communications, Vol 31, Iss 10, pp. 2142-2149, Jun 2008.

[Yan03] Yang S., Su S., Lam H., “A Non-Repudiation Message Transfer Protocol for E-commerce”, Proceedings of IEEE International Conference on E-Commerce Technology, Jun 2003.

[Yee04] Yee K., "Aligning Security and Usability", IEEE Security & Privacy Magazine, Vol. 2, Iss 5, pp. 48-55, Sep 2004.

[Yeg11] Yegin A., Ohba Y., Morand L., Kaippallimalil J., ”Protocol for Carrying Authentication for Network Access (PANA) with IPv4 Unspecified Address” Internet Draft draft-yegin-pana-unspecified-addr-04 (work in progress), Mar 2011.

[Ylo96] Ylönen T., “SSH - Secure Login Connections over the Internet”, Proceedings of Sixth USENIX UNIX Security Symposium, Jul 1996.

[Zho96a] Zhou J., Gollmann D., “A fair non-repudiation protocol”, Proceedings of IEEE Symposium on Security and Privacy, May 1996.

[Zho96b] Zhou J., Gollmann D., “Observations on Non-repudiation”, Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology, Nov 1996.

[Zho97a] Zhou J., Gollmann D., “An Efficient Non-Repudiation Protocol”, Proceedings of the 10th Computer Security Foundations Workshop, Jun 1997.

[Zho97b] Zhou J., Gollmann D., “Evidence and non-repudiation”, Journal of Network and Computer Applications, Vol 20, Iss 3, pp. 267-281, Jul 1997.

[Zho98] Zhou J., Lam. K. “Undeniable Billing in Mobile Communication”, Proceedings of 4th ACM/IEEE International Conference on Mobile Computing and Networking, Oct 1998.

[Zho99a] Zhou J., Lam K.Y. “A Secure Pay-per View Scheme for Web-Based Video Service”, Proceedings of the Second International Workshop on Practice and Theory in Public Key Cryptography, Mar 1999.

[Zho99b] Zhou J., Lam K.Y., “Securing digital signatures for non-repudiation”, Computer Communications, Vol 22, Iss 8, pp. 710-716, May 1999.

[Zim09] Zimmermann P., Callas J., ”The Evolution of PGP’s Web of Trust”, appearing in Beautiful Security by Oram A., Viega J. (Eds.), O’Reilly Media, 2009.

PUBLICATIONS

Publication P1

Heikkinen S., Tschofenig H., "HIP Based Approach for Configuration Provisioning", in *Proceedings of The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, Helsinki, Finland, Sep 2006.

The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)

HIP BASED APPROACH FOR CONFIGURATION PROVISIONING

Seppo Heikkinen
Tampere University of Technology
Tampere, Finland

Hannes Tschofenig
Siemens AG
Munich, Germany

ABSTRACT

The most typical configuration procedure of a host involves the provision of an IP address and most often this is done with the help of Dynamic Host Configuration Protocol (DHCP). Unfortunately, the security of this procedure is largely non-existent. While the closed nature of the access networks has mitigated the vulnerability, the evolvement of the networks and increase in wireless use demand more stringent secure measures. This paper proposes the integration of DHCP with Host Identity Protocol (HIP) mechanisms, so that the security measures inherent to HIP can be extended to protect the configuration information and its provisioning as well.

I. INTRODUCTION

In IP communication the host quite naturally needs an IP address before it is able to communicate with other hosts. It also needs some additional network information before being able to establish a full fledged communication with different kind of services. The networks themselves are likely to evolve past the static autonomous entities they are today and it is likely that the networks will engage in cooperation and share resources in ways that are not currently practical. This kind of setting will require new types of configuration information that will be exchanged between the entities, which can be not just individual nodes but whole networks. Consequently, the limited security that is provided today is not enough, especially if shared medium is used. In addition, the different entities require clear naming in order to be able to identify the party they are going to be interacting with. In other words, having a verifiable identity, such as those based on cryptographic identifiers, will make the design of security solutions easier.

In this paper we will present a new way of protecting the exchange of configuration information between two entities. We will discuss the approach in relation to two communicating nodes, but they could be seen as representatives of their respective networks. They could be functioning under their own identities, but they also could be using the identity of the whole network, if such privilege was granted to them. The proposal addresses the security shortcomings of the current configuration protocols, namely those evident in the deployment of Dynamic Host Configuration Protocol (DHCP). The emphasis is mainly, with respect to the practicality of the current networks, in the configuration of IP connectivity. The phases that precede the configuration step are assumed to be handled by some other means. In other words, it is assumed that link layer connectivity and neighbour discovery has already taken place. Thus, our proposal investigates the advantages of employing Host Identity Protocol (HIP) in combination with DHCP to protect the configuration exchange between the entities, which are identified by their public keys. The motivation of investigating the benefits of HIP stem from its potential to become the identity layer mechanism for the future networks and as such it would be included in every communication exchange taking place between two peers.

The paper is organised as follows. The next section discusses the related work in configuration provisioning. After that we give a high level presentation of our proposal. In the fourth section we go deeper into the details and discuss the possibility to use indirection architecture to route the configuration information. The fifth section presents authorisation aspects that need to be taken into consideration to ensure the legitimacy of actions. The sixth section concludes our paper.

II. RELATED WORK

DHCP contains security features to ensure the authenticity of the messages, although they do not address denial of service (DoS) concerns [1]. Additionally, the security measures assume out of band mechanisms, such as manual configuration, for key management, which clearly does not scale well and has been the main reason preventing the deployment. Specifications also assume that configuration information does not require confidentiality protection, even though DHCPv6 transmits reconfiguration key information between the parties [2].

Many existing access networks rely on the link layer security when securing the configuration provisioning. This includes, for example, WLAN and UMTS networks, which are assumed to be trustworthy on the infrastructure side. While it is typically true that the nodes in the infrastructure belong to the same administrative domain, it does not necessarily mean that the domain itself is trustworthy. Even in UMTS, which supposedly provides mutual authentication, the client actually does not know the identity of the access network, but it has to assume that the access network has acquired the authentication vector legitimately [3]. Thus, the networks may use unprotected DHCP as their configuration providing mechanism. It is employed in a similar fashion in schemes involving higher layers. Such schemes can be, for example, Web portals, where the client uses a credit card to pay for the WLAN usage. This can lead to the disclosure of the credit card number to the unauthorised parties, though, if the communication to the server is not protected adequately.

There exists various network level schemes for authenticating network access, such as PANA [4], but they typically require that the client already is in the possession of IP level connectivity. This can, of course, include 802.1X type of controlled access [5] or any other filtering mechanism that ensures that no additional communication is possible, before the authentication has succeeded, or the provided preliminary address is very short lived. Also, the access point controlling the access can receive configuration information concerning

the client, like IP address, through mechanisms like RADIUS or Diameter. Such frameworks can also use Extensible Authentication Protocol (EAP) to authenticate the client in various ways [6]. Almost all EAP methods provide the ability to export keying material as part of successful EAP method protocol run [7]. These frameworks conceptually usually involve three entities, i.e., the EAP peer, the Authenticator, and the EAP server. IKEv2 also operates on network layer and is able to provide the client with a limited set of configuration parameters, but it still requires a preliminary address before engaging itself in IKEv2 negotiation [8].

In addition to the rather limited security measures of the DHCP specification there exist some suggestions for enhancing DHCP security. Some proposals suggest signing of messages with a private key, for which the public key could be found in DNS [9]. Another approach uses certificates to provide authenticity and authorisation [10] and uses a binary encoding [11] for compression of the certificates in order to make them fit, although slightly violating the original specification. IPsec protection of DHCP messages is provided in [12], but the usage scenario relates to the tunnelled remote access rather than providing first level connectivity. EAP is also suggested as one potential option for providing a keying framework for DHCP [13].

The concept of employing HIP with DHCP already appeared in [14], but it only sketched the concept on a very high level and did not yet pay attention to the details. A slightly more drastic approach was taken in [15], which, even though employing similar kind of principles, defined a whole new architecture for network attachment.

III. HIGH LEVEL OVERVIEW

The basic idea is to carry configuration information along with the protocol execution of HIP, i.e., the so called base exchange [16]. In other words, as the two entities, identified by their public keys, form a security association between each other with the help of Diffie-Hellman exchange, at the same time they also exchange configuration information that can be used, for example, to provide an address to the Initiator. This enables the configuration information to take advantage of the protection provided by the HIP mechanisms, thus ensuring the integrity of data. Additionally, it is possible to encrypt information and include authorisation tokens into the messages, so that the legitimacy of the request can be assured. So, in essence, with the four message handshake the parties are able to establish a security association and exchange basic configuration for connectivity. Thus, the amount of messages for gaining basic connectivity is minimised, even though computationally the procedure is more demanding than compared to, for example, the basic DHCP exchange. To summarise, the protocol consists of two phases: secure channel establishment and authorisation information provision. The secure channel is established either through mutual or unilateral authentication, but it could also be completely anonymous. After the base exchange the parties can continue to exchange any data and they can use the security association to protect it with IPsec. This could be, for example, QoS signalling or it could be used as an enabler for access control.

We could further note that this procedure could be used between networks, as well. In other words, the two parties could be representatives of their respective networks and the configuration information would relate to their parameters. This could be, for example, network prefix delegation [17]. But we further want to note that even though there exist initiatives for network mobility, like NEMO working group in IETF, we expect that this scenario is targeted better to the future networks, such as those proposed by the Ambient Networks project [18].

The protocol run starts with the I1 message, which is used to discover the relevant entity (see Fig. 1). The Initiator may know the host identity representation, Host Identity Tag (HIT), of the Responder, if it has learnt it during the neighbour discovery, for instance, but it may leave the target identifier empty and only seek for the closest server providing DHCP service, in a similar fashion as anycast addresses are intended to be used. Access points that do not include co-located configuration server can work as relays and forward the messages to the relevant servers. This can take advantage of indirection architecture, like the one proposed in [19]. The first message could already contain a configuration request, but for the sake of avoiding DoS the Responder might not want to react to it, especially if it entailed reserving some resources.

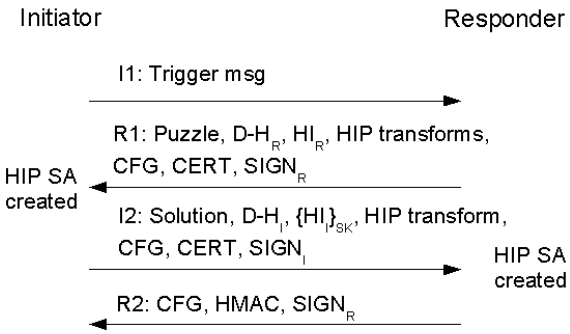


Figure 1: Enhanced HIP exchange

The Responder responds with R1 that contains the typical HIP components. It can also contain configuration information that does not reserve any state, for example announcement of network prefixes. Of course, if the site policy so dictates, it might be possible to include some address configuration already in this packet, but it would not be integrity protected by the signature as this would defeat the idea of using precalculated signatures with R1 packets. The Responder can also include certificates, which state its authority to assign certain addresses through this particular access point. They could also make statements about the authenticity of the Responder identity from the point of view of some trusted third party.

With I2 message the Initiator is able to prove its legitimacy or at least show that it has expended some effort to compute the provided puzzle. At this point it also is able to calculate

common key material and use that to protect the configuration information as well. Additionally, the Initiator can include a certificate or other security token that provides authenticity to the used host identity. With that one might attach an authorisation statement that gives credibility to the request, i.e., a third party might vouch for the identity to be authorised to receive configuration or it could be some other form of promise of compensation. The authorisation statement could very well be linked to an ephemeral identity. In other words, the Initiator in essence remains anonymous, but is able to present a statement by a third party that is enough to classify it as a trustworthy peer for communication.

The R2 contains the response to the configuration request and typically contains addressing information that the Initiator can use to configure its IP stack. The provided address could be a Cryptographically Generated Address (CGA) [20], so that the client is able prove the ownership of the address during a later communication with some other entity. This information is protected by using the normal HIP mechanisms. Note that if the Responder did not provide any authorisation statement about the provisioned addresses in its R1 message, it should do it now, otherwise the Initiator cannot be sure about the legitimacy of the configuration it receives. This could be linked to any authorisations that the access point has potentially already provided or vice versa.

If the parties need to exchange any updated configuration or additional security tokens, they can use HIP UPDATE packets to do that, which can also be used to update the relevant security associations. In the end the Initiator can relinquish its addresses with the HIP CLOSE packet, which also closes the HIP association. Typical DHCP timeout approach for address leases can also be used, in case the connection is not torn down gracefully. The lease can be updated with the aforementioned UPDATE packet.

IV. DISCOVERING SERVERS

One of the central ideas of Ambient Networks is the flexible and dynamic nature of establishing networks with a constant flow of nodes joining and leaving. These nodes contribute to the functionality offered by a network. With today's network this configuration is typically kept in a central database either co-located with the DHCP server or allowing the DHCP server to access the database. In our architecture we assume a large number of these servers to be available possibly keeping only a subset of the available information.

To show the different architecture we envision we first explain how DHCPv6 allows DHCP clients to find DHCP servers. DHCP client transmits a message to a special multicast address, All_DHCP_Servers (FF05::1:3) [21]. At this point it is presumed to be in the possession of a link local address for its interface. If the DHCP server happens to be on the same link as the client, the server receives this message and can reply directly to the local address. It is, however, possible that the server is not on the same link. Then it is the responsibility of a DHCP relay agent to forward the messages. They might know the server address and send the message directly to it, but if they do not know it, then they have to forward the wrapped message onwards to other relay agents. When the server replies to the original request, it will go back through the same chain of relay agents by using unicast.

In our approach the case, where the server is on the same link, is straightforward, but in case the server is behind one or more relay agents then the above procedure is more difficult to deploy. Of course, if the first relay knows the server, it can always tunnel the whole packet. After all, the signalling traffic between relays and servers is expected to be protected with IPsec [21]. In this case, the inner addresses of the tunnelled IPsec packets could be HITs. In case the server is not known the multicast mechanism ought to be used and the original HIP packet wrapped inside a new packet.

There are, however, a few alternative deployment models that need to be analysed. The first obvious consideration would relate to the HIP rendezvous mechanisms, where a relay server is used to convey the initial exchange between HIP peers [22]. The DHCP server would be expected to be registered to the rendezvous server beforehand, so the messages could be forwarded to the correct entity. Unfortunately, this would again introduce a single-point of failure even though the rendezvous server only needs to see the I1 message. There are suggestions concerning two-way forwarding, though [23]. We believe that a better alternative is to consider an indirection architecture, like the one proposed with Hi^3 [19].

The Host Identity Indirection Infrastructure (Hi^3) is an architecture that combines the ideas of HIP and Secure-i^3, which is basically an overlay connectivity and routing architecture with support for mobility [24]. To some extent one might view it as identity layer routing. The idea is to have triggers within the infrastructure that can be used to contact the destination identified by its destination identifier. In other words, when a host sends data to the target host, the triggers in the infrastructure catch this, resolve it to the current location, and deliver data to the correct host. The control information is stored within the infrastructure using some distributed approach, such as Distributed Hash Tables (DHT) [25]. So, the hosts do not need to be identified by their IP addresses, but it is sufficient to send data to the identifiers that are represented by the HITs. The hosts can modify their triggers to map to their currently used locator (or to another trigger), but from the sender's point of view this is completely transparent. Triggers can be public or private depending whether they are used in the initial contact or in a short term communication and basically they could form chains, i.e., the trigger could point to another trigger instead of a location. In the Hi^3 world the triggers are intended to be used to convey HIP control packets, whereas the regular traffic could still use the IP connectivity protected with IPsec, although they could take advantage of the IPsec aware middle-boxes, SPINATs, for the purposes of routing and additional protection against DDoS [26]. In essence, the introduction of Hi^3 would form a secure control plane for the Internet [19].

In our case, DHCP servers are registered at the DHT using an anycast trigger. The DHCP relay would use the anycast trigger, which would be publicly known, to point to the nearest DHCP server or servers. Hence, the client would send its initial I1 to this trigger, which would then get delivered to

the closest DHCP server by the infrastructure. At the same time the private trigger of the end host would get registered by the access point on behalf of the end host in order to allow the server to send its replies to it. If multiple DHCP servers are registered to this anycast trigger, then the client receives several advertisements in the form of R1s. The client has then the option of choosing with which server or servers to proceed with the base exchange as in distributed approach different servers could contain different subsets of network information. There is also a suggestion in [27] that the target HIT of I1 could be a sort of a service identifier, i.e., public value, and the server subsequently would use its own private HIT in communication with the client. This would also require delegation between those two different HITs, otherwise the client could not trust the changed HIT of its presumed peer [28]. The service identifier could be advertised, for example, by the access point. Hi^3 additionally suggests a feature, which would enable the infrastructure to respond to the first I1. The Responder needs to provide the infrastructure with precomputed R1s, which the infrastructure can use to reply to I1s. It is even possible that the infrastructure validates the puzzle solution in I2 before delivering it to the Responder. This further enhances the DoS resistance. Care must be taken when implementing the trigger registration, though, otherwise it might be possible to execute similar kind of traffic rerouting attacks as in Mobile IP. Cryptographic properties of HITs can be employed to alleviate this concern. An important point to realise with this proposed scheme is that the client actually does not have to be aware of Hi^3, because the access network infrastructure takes care of the routing and other specifics. Of course, if the client were a fast moving mobile, then it certainly would benefit from the mobility functionality.

As a final remark, it can be stated that even though Hi^3 provides a secure and robust infrastructure, in the short term DHCP relays are more likely to be configured with HITs of their DHCP servers before a fully dynamic infrastructure is used. Some performance measurements of the indirection infrastructure (including latency analysis) are provided in [27], although one could expect that the access network does not contain as many nodes as investigated in [27].

V. AUTHORISATION ASPECTS

While using basic HIP mechanisms already provides a certain level of protection, i.e. the parties are authenticated opportunistically, additional level of security can be provided with separate authentication and authorisation statements. Authentication statements, such as certificates, can give credibility to the used identifiers, i.e. they belong to some known identity, but it is more important to have authorisation statements, which provide assurance that the entity is performing a legitimate action. Authorisation statements bring more policy granularity and they can be bound to identifiers, which only have limited lifetime. This approach has privacy preserving qualities.

Authorisation statements can be made with techniques such as Security Assertion Markup Language (SAML) and certificates. SAML is an XML based language for conveying assertions regarding the characteristics of an entity, which usually are protected by cryptographic means, e.g., signatures [29]. Certificates, on the other hand, come in many different flavours. Perhaps the best known example is X.509 certificate, although it is more suited for entity authentication. Better alternatives for our purposes are Simple PKI (SPKI) [30] and attribute certificates [31], which are more purpose oriented and provide means to bind a key to rights or attributes.

An extra complication in our approach is caused by length restrictions. HIP allows 2008 bytes to be used for the parameters inside the HIP packet and the mandatory parameters of the base exchange decrease this even further. Additionally, we want to include DHCP options within the same space. The most "fully packed" HIP packets are R1 and I2, which in the worst case can consume well over 1000 bytes, although large part of this is due to the Host Identity, which can include a lengthy domain identifier. DHCP options typically consume additional 300-400 bytes, which leaves very little space for any authorisation statements or tokens. For example, neighbour discovery security mechanism test runs report certificate lengths between 864 and 888 [32]. Clearly, we are not able to include lengthy certificate chains within the messages, but some statements can still be made. Encoded SPKI certificates used in [10] consumed around 250 bytes, which could be used to signal the authority of the server to provide configurations, for instance. It is also possible to use SAML with the help of Artifacts, i.e. a 44 bytes long construct that is used to reference the SAML Assertion. While this saves a lot of space, it requires existing connectivity so that the actual assertion can be fetched from the referenced location. If the client is in the process of configuring its connectivity, the scheme is more suited for the server, which can fetch and check the presented Artifact. Of course, it is possible that the client is granted limited connectivity for accessing external servers, but this opens an additional possibility for abuse.

Note that the authorisation statements only make sense, if the third party giving them is trusted by both communicating parties or they are both able to construct trust paths leading to the same third party As mentioned above, long chains leading to trusted keys do not fit in the messages, so the chains have to be constructed by other means. It is possible that the parties are in possession of the trusted public keys through preconfiguration or the relevant chains could be learnt during the neighbour discovery [32]. Additional messages can also be exchanged after the base exchange, even though this increases the total number of roundtrips needed for establishing connectivity.

VI. CONCLUSIONS

In this paper we proposed a scheme for integrating HIP and DHCP. While the large scale deployment of HIP is not yet certain, the proposal is able to natively provide confidentiality, integrity, and key management services. In addition, the involved parties are clearly named with help of their public keys and masquerading would require knowledge of the corresponding private key. Even though the

confidentiality of typical DHCP configuration may not have much value currently, the evolving network scenarios might prove differently in the future. The proposal also helps to mitigate the denial of service attacks against the server. To some extent, the potential use of indirection infrastructure can provide additional protection against flooding attack both for the client and the server. Man-in-the-Middle attacks are a concern, but they can be prevented with the use of security tokens provided by a trusted third party. This includes authorisation information, which is suited for ephemeral identifiers in privacy sensitive scenarios. Explicit authorisation also brings more granularity to the rights management, thus granting only the required privileges, i.e., the so called least privilege principle. The use of authorisation tokens may require additional message exchanges, though, and therefore have performance impact, unless more drastic changes to the basic HIP draft are devised.

VII. ACKNOWLEDGEMENT

The authors had the ideas of the proposal for the first time within the EU funded Ambient Networks project and wish to acknowledge the feedback and support received there. Also, the authors wish to thank the anonymous reviewers for their comments.

REFERENCES

[1] Droms. R. (Ed.), Arbaugh W. (Ed.). Authentication for DHCP Messages. IETF RFC 3118. Jun 2001.

[2] Droms R. (Ed.). Dynamic Host Configuration Protocol for IPv6 (DHCPv6). IETF RFC 3315. Jul 2003.

[3] 3rd Generation Partnership Project. 3G Security, Security Architecture (Release 6). 3GPP TS 33.102. Dec 2005.

[4] Forsberg D., Ohba Y. (Ed.), Patil P., Tschofenig H.., Yegin A. Protoocol for Carrying Authentication for Network Access (PANA). IETF Internet-Draft, Work in progress. Mar 2006.

[5] IEEE. Port-Based Network Access Control. IEEE Std 802.1X-2001. Oct 2001.

[6] Aboba B., Blunk L., Vollbrech J., Carlson J., Levkowetz H. (Ed.). Extensible Authentication Protocol (EAP). IETF RFC 3748. Jun 2004.

[7] Aboba B., Simon D., Arkko J., Eronen P., Levkowetz H. (Ed.). Extensible Authentication Protocol (EAP) Key Management Framework. IETF Internet-Draft, Work in progress. Jan 2006.

[8] Kaufman C. (Ed.). Internet Key Exchange (IKEv2) Protocol. IETF RFC 4306. Dec 2005.

[9] Lemon T., Richardson M. DHCP RSA/DSA Authentication using DNS KEY records. IETF Internet-Draft, Expired. Jun 2003.

[10] Arbaugh W., Keromytis A., Smith J. DHCP++: Applying an efficient implementation method for fail-stop cryptographic protocols. Proceedings of Global Internet, GlobeComm '98.Aug 1998.

[11] Arbaugh W., Keromytis A., Farber D., Smith J. Automated Recovery in a Secure Bootstrap Process. Internet Society 1998 Symposium on Network and Distributed System Security. Mar 1998.

[12] Patel B., Aboba B., Kelly S., Gupta V. Dynamic Host Configuration Protocol (DHCPv4) Configuration of IPsec Tunnel Mode. IETF RFC 3456. Jan 2003.

[13] Tschofenig H., Yegin A., Forsberg D. Bootstrapping RFC3118 Delayed DHCP Authentication using EAP-based Network Access Authentication. IETF Internet-Draft, Work in progress. Feb 2006.

[14] Heikkinen S., Tschofenig H., Gelbord B. Network Attachment and Address Configuration using HIP. Position paper, Workshop on HIP and related architectures, Washington D.C. Nov 2004.

[15] Arkko J., Eronen P., Tschofenig H., Heikkinen S., Prasad A. Quick NAP - Secure and Efficient Network Access Protocol. 6th International Workshop on Applications and Services in Wireless Networks (to appear). May 2006.

[16] Moskowitz R., Nikander P., Jokela P. (Ed.), Henderson T. Host Identity Protocol. IETF Internet-Draft, Work in progress. Oct 2005.

[17] Troan O., Droms R. IPv6 Prefix Option for Dynamic Host Configuration Protocol (DHCP) version 6. IETF RFC 3633. Dec 2003.

[18] EU FP6 IST Ambient Networks project, http://www.ambient-networks.org/

[19] Nikander P., Arkko J., Ohlman B. Host Identity Indirection Infrastructure (Hi3). Proceedings of Swedish National Computer Networking Workshop (SNCNW) 2004. Nov 2004.

[20] Aura T. Cryptographically Generated Addresses (CGA). IETF RFC 3972. Mar 2005.

[21] Droms R. (Ed.). Dynamic Host Configuration Protocol for IPv6 (DHCPv6). IETF RFC 3315. Jul 2003.

[22] Laganier J., Eggert L. Host Identity Protocol (HIP) Rendezvous Extension. IETF Internet-Draft, Work in progress. Oct 2005.

[23] Nikander P., Ylitalo J., Wall J. Integrating Security, Mobility and Multi-homing in a HIP Way. Proceedings of Network and Distributed Systems Security Symposium (NDSS'03). Feb 2003.

[24] Adkins D., Lakshminarayanan K., Perrig A., Stoica I. Towards a More Functional and Secure Network Infrastructure. Technical Report UCB/CSD-03-1242, Computer Science Division (EECS), University of California, Berkeley. 2003.

[25] Stoica I., Morris R., Karger D., Kaashoek M., Balakrishnan H. Chord: A Scalable Peer-to-peer Lookup Service for Internet Application. Proceedings of the 2001 ACM SIGCOMM. Aug 2001.

[26] Ylitalo J., Melen J., Nikander P., Torvinen V. Re-thinking Security in IP based Micro-Mobility. Proceedings of 7th Information Security Conference (ISC04). Feb 2004.

[27] Gurtov A., Korzun D., Nikander P. Hi3: An Efficient and Secure Networking Architecture for Mobile Hosts. HIIT Technical Report 2005-02. Jun 2005.

[28] Koponen T., Gurtov A., Nikander P. Application mobility with HIP. Proceedings of NDSS Wireless and Mobile Security Workshop. Feb 2005.

[29] Cantor S. (Ed.), Kemp J. (Ed.), Philpott R. (Ed.), Maler E. (Ed.). Assertions and Protocols for the OASIS Security and Assertion Markup Language (SAML) V2.0. OASIS Standard. Mar 2005.

[30] Ellison C. (Ed.). SPKI Certificate Theory. IETF RFC 2693. Sep 1999.

[31] Farrell S., Housley R. An Internet Attribute Certificate Profile for Authorization. IETF RFC 3281. Apr 2002.

[32] Arkko J. (Ed.), Kempf J., Zill B., Nikander P. SEcure Neighbor Discovery (SEND). IETF RFC 3971. Mar 2005.

Publication P2

Heikkinen S., "Authorising HIP enabled communication", in *Proceedings of The 10th International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS07)*, San Diego, USA, Jul 2007.

Publication P3

Heikkinen S., "Applicability of Host Identities to Implement Non-Repudiable Service Usage", in *International Journal On Advances in Systems and Measurements*, vol. 1, nr. 1, pp. 14-28, 2008.

Applicability of Host Identities to Implement Non-Repudiable Service Usage

Seppo Heikkinen
Tampere University of Technology
firstname.lastname@tut.fi

Abstract

In a typical roaming scenario the accounting information received from the roaming partner is expected to be trustworthy. Things like fear of losing one's reputation have been working as disincentives for fraudulent behaviour between the large operators. However, when smaller players enter the market and steps are taken towards more dynamic relationships as in the visions of ubiquitous computing environments, the need for reliable records becomes paramount. Thus, secure accounting mechanisms are needed for ensuring correct compensation amongst the interoperating partners. On top of that, the partners need to be authorised with sufficient granularity to be able to engage in the transaction in the first place. The mere authentication should not be enough.

In this article we present a solution concept for ensuring non-repudiation of the service usage, so that cryptographically secure accounting records can be generated, and the parties involved in the transaction make their commitments only to the resources actually consumed. The solution is based on the employment of Host Identity Protocol (HIP) and hash chains, so that we can provide a convenient binding between the identity and authorisation information. Also, in order to avoid service hijacking, mechanisms for binding this information to the actual traffic are discussed.

Keywords: hash chains, host identity, non-repudiation, service

1. Introduction

The communication environment is changing. As the ubiquitous computing paradigms gain more momentum and the technological development allows more dynamic usage patterns and relationships, more and more small players enter the market to get their piece of the service provisioning cake. Naturally, these players want to ensure that they receive authentic users that are able to pay for the service usage. On the other hand, the players vouching for the liability of the users want to make sure that the generated expenses are within certain limits, i.e., they want to control how much risk they are willing to take on behalf of their customers. This requires measures to ensure the correct authorisation for the users of the systems.

Thus, we have service providers, who want to receive compensation for the provision of their service resources. They are complemented by the third parties, such as home operators, who help in authenticating users and ensuring that the generated costs will be covered. Finally, we have the users, who want to make sure that they receive the service that is promised and that it is correctly charged. After all, the appearance of unauthorised charges on phone bills, i.e., cramming, is not unheard of amongst the consumers [1]. Sometimes, the user may not even have clear notion about the identity of the responsible service provider, as is often the case with visited access networks, even though the access network might be in the possession of authentication material generated by the home network.

As the interaction and the established relationships are more dynamic in nature and lasting perhaps only one transaction, typical assumption that the loss of reputation is incentive to ensure the correctness of accounting records is no longer valid. Hence, we need mechanisms that create secure accounting records so that the service transaction is undeniable and authentic for the both parties of the transaction. We propose such a simple non-repudiable mechanism that takes advantage of Host Identity Protocol (HIP) and hash chains. Our focus is on the interaction of the user and the service, not so much in the negotiation between the service and the third party nor the bootstrapping of trust between the user and the third party.

HIP already provides end point authentication and simple key exchange, but it does not currently address the problem of authorisation to the sufficient detail. In order to implement the suggested Non-Repudiable Service Usage (NoRSU), one point of this article is to

discuss how to include authorisation tokens into HIP and what are the consequences. Additionally, the hash chains are employed to introduce an incremental payment solution, i.e., a chain of tokens is created by repeatedly hashing a secret seed value. Thus, the service provider is able to generate undeniable charging records and the user can be sure that the charging is based on actual use. As HIP assigns cryptographic identities to the communication end points, the tokens can be tightly bound to the actual communication. HIP also introduces a handshake procedure for negotiating and establishing a security association between the end points. For the benefit of performance this procedure can be overloaded with the compensation related information. Thus, no additional roundtrips are introduced.

This article is organised as follows. The next section discusses the related work. The third section describes the details of the proposed system and the section after that gives examples of two use cases. The fifth section discusses the limitations the implementations have to take into account and suggests ways to efficiently encode the used information in order to overcome these limitations. The sixth section analyses the solution in terms of threats that can be faced. Finally, the seventh section concludes the article.

2. Related work

HIP is an experimental proposal for future network architectures that introduces a new identity layer between the network and transport layers [2]. This allows decoupling the dual role of the IP addresses. That is, currently they function as end point identities and locators. In the HIP model the end points are identified by their cryptographic identifiers, called Host Identity Tags (HIT), which are derived from their public keys. This accommodates for end host authentication and simple key exchange. Thus, the parties are able to setup a security association between themselves, which can be used to protect the control information exchange. Additionally, the protection of subsequent data transport is possible with IPsec ESP [3]. Other transports can be defined, too.

HIP uses four messages in the so called base exchange to establish the identity of the parties and to create the needed keying material with the help of Diffie-Hellman key exchange (see Figure 1). For the purposes of the paper the initiator and the responder can be considered as the client and the server, respectively. Besides securing the message exchange, the protocol mitigates denial of service (DoS) attacks by introducing a puzzle scheme.

An initial proposal for including authorisation to HIP has been introduced, but that work is still very much in the draft stage and basically provides a placeholder for the certificates [4]. Like the proposal, [5] and [6] also discuss the possibility of including Simple Public Key Infrastructure (SPKI) certificates in the protocol, but do not analyse the use case thoroughly, even though [5] provides a prototype implementation adapted to grid environments. There is also a general sketch of an attachment architecture, which includes both HIP and compensation related issues in [7]. A solution employing hash chains and KeyNote credentials to implement One Time Password (OTP) coins was depicted in [8], even though without clear binding to the actual communication. Similar ideas were used to sketch a high level solution presented in [9], but it used SPKI certificates instead of KeyNote and already took advantage of HIP to ensure the binding to the actual traffic. The text presented here extends that work with additional details.

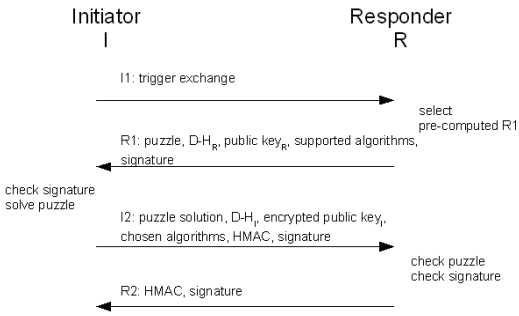


Figure 1. HIP base exchange

Hash chains have been used for password solutions and such one-time password authentication was suggested already in 1981 by Lamport [10]. The idea of hash chains is based on the irreversible nature of the hash functions. In other words, you are not able to calculate the source value once you have the result of the function. Hash chain is created by applying a secure hash function in successive fashion to a secret seed value and then using the values of the hash calculations in reverse order. So, it is very easy to check by applying one hash operation to the previously received value that the current value is part of the chain, but very hard to calculate additional values without the knowledge of the initial seed value of the chain. The idea behind hash chains is illustrated in Figure 2.

There exists also several other works, which have considered employing hash chains to introduce non-repudiable billing and micropayments in various

scenarios, so the concept is not new. [11] uses hash chains to implement a payment solution for ad hoc networks, but it requires the use of smart card technology to control the release of hash chain values. [12] also presents a protocol for undeniable billing with entity authentication and privacy support for mobile networks roaming access using hash chains, although it requires online interaction with the home network.

x = secret seed value

H = Hash function

n = length of chain

$H^n(x) = H(H^{n-1}(x))$

$H^n(x)$ = hash chain anchor

release chain values:

$H^n(x)$, $H^{n-1}(x)$, $H^{n-2}(x)$, ...

Figure 2. Idea behind hash chains

3. HIP based non-repudiation

This section describes how the hash chains are integrated with the HIP base exchange.

3.1 General overview

Our proposal, which is based on the aforementioned HIP, works in the way depicted in Figure 3 (HIP specific parameters left out). The basic idea is to add extra information to the HIP messages in order to negotiate the usage of non-repudiative accounting within the communication. So, in a sense, we are negotiating a non-repudiation association in addition to the identity association.

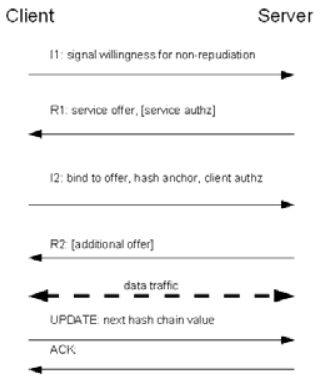


Figure 3. Base exchange with non-repudiation enhancements

A new HIP parameter is needed to signal the intent to access certain service with the capability of using NoRSU. Note that the server could also send this kind of indication to the client at some later point using the HIP UPDATE packets along with the corresponding offer, if the client tried to access a service, for which the server required extra accounting (provided they had an existing HIP association). Figure 4 gives an example of the said HIP parameter for indicating the use of non-repudiation for certain service and it also shows the general Type-Length-Value (TLV) format of HIP parameters (C bit denotes possible critical parameter).

Type (15 bits)		C	Length (16 bits)
Subtype of NoRSU	Encoding of name		Service name length (16 bits)
Service name with the indicated encoding + padding if needed (variable length)			

Figure 4. New HIP parameter for signalling the use of non-repudiation for a specific service

3.2. Modified base exchange messages

The tasks for individual HIP messages are as follows. The I1 message functions as a trigger message as in basic HIP base exchange [2], but with the addition of possibility to signal the capability of the client to engage in a NoRSU exchange as discussed above. The server's response, R1, contains an offer in the form of an SPKI certificate for the usage parameters, including the number of tokens needed for certain amount of time or byte count, e.g., you need one token per minute or you need one token per 100 kilobytes. That is really up to the charging scheme of the provider, but generally the value of one token should be kept small in order to avoid big losses in case of abuse. There could also be an additional advice of charge functionality telling the value of one hash chain token in monetary terms. However, this does not take into account the possibility that the tariffs are different between different parties, i.e., some client have a "better deal", because they are subscribers of some favoured organisation, for instance.

The offer is protected by a signature, which also binds the provider to the offer. The signature could also be made by a trusted third party (TTP) in order to guarantee that the server is a legitimate one, but this could also be done by using an additional authorisation certificate (see subsection 3.3). The latter case is more flexible and gives more control of the tariffs to the

service provider, naturally within the limits of the third party authorisation.

Offer should also contain validity date, so that the provider has better control of the expiration of the used offers. This allows, for example, using different tariffs at different times of the day. Naturally, if the session continues after the validity period the parties should renew their contract provided the new offer is satisfactory. It is the responsibility of the client to make sure that no additional hash chain values are sent with the assumption that the old offer is still valid. The server can in this case just stop serving the client, if there is no response to the new offer.

Note that instead of offering certain time or rate based traffic the offer could just be for the use of certain service, which could be described by a profile or a service name. This naturally requires that there is common consensus about the semantics of such profiles, but that can be agreed when establishing trust relationship, e.g., a roaming agreement, with the third party. There could also be several offers, e.g., choice between time and byte count, but this has restrictions as space is limited (see section 5).

If the offer meets the requirements of the client, it sends a response in the I2 message, which contains the signed acceptance of the offer. The acceptance is indicated by calculating a hash over the offer and signing it. Additionally, the response must contain the hash anchor value, which the server can use to validate the subsequent values, i.e., it acts as the starting point for the hash chain, which the client has created. It can also be used to identify the whole hash chain among several parallel chains.

The fourth message of the exchange, i.e., R2, can just acknowledge the validity of the offering process and, for instance, show as a summary what kind of agreement is in effect. Some advanced scenarios are possible, though. One could relate to special offers, i.e., if the customers of certain operator were allowed to get even cheaper service, R2 could contain a special offer with a reduced tariff. This could mean, for example, a longer interval between subsequent hash chain values. The client would need to send an additional control message to sign the offer with a new hash chain anchor value. Otherwise, it could still be charged the higher price. This could be found out, though, when (or if) the client disputes the costs and presents the alternative offer. So, in low value transactions it could be possible to just use R2 to signal that the hash chain value interval is shifted (for the benefit of the client). Another approach is that the value of token is lower between service provider and the third party, thus the generated bill is lower.

3.3. Authorisation issues

While the service provider might have some external knowledge about the client's liability for service usage based on its identity, generally the client also needs to attach an authorisation statement from TTP that states that the client is trustworthy to receive the specified service for the specified amount. The service might be specified based on service types or it could be specified on the provider level. In other words, the specified service and the service offer identities should then match. This is a slightly less flexible option, but provides more security, because overspending can be controlled more easily. In case the service granularity is just based on the service type, the client could use several different service providers for the maximum amount defined in the certificate during the validity period. Of course, at the time of the clearing the third party would notice this and could initiate appropriate procedures against the client. This is really no different from the way post-paid phone calls are charged. Thus, TTP has the liability, but it can still control what sort of certificates it issues and hence manage its own customer risk. Issuing only short lived authorisations is also one way of mitigating risk.

```
(
 (cert
  (subject (hash sha1 <hash value>))
  (issuer (hash sha1 <hash value>))
  (target-service-url (hash sha1 <hash value>)
  (amount-max (time (s 3600)))
  (propagate)
  (validity
   (not-before 2008-07-29_12:00:00)
   (not-after 2008-07-30_12:00:00))
)
 (signature (dsa-sha1 <sig>))
)
```

Figure 5. Example of TTP certificate

In Figure 5 we give an example of a third party authorisation in the form of an SPKI certificate, which allows certain subject to access the indicated service. The subject is identified by the hash of the public key, even though one could also use HIT to identify the party. However, as HIT includes IPv6 kind of interpretation (see [13]), there is slightly larger chance of collision than in the case of hashing a public key. The target service is also indicated with the hash value calculated from the service URL (or just with suitable URN) and the maximum service time is also indicated in order to limit the "credit" of the client. Note that

TTP certificates could authorise various other things as well and act as a policy distribution mechanism. This is really up to the agreement made by the service provider and the third party.

The example certificate also includes the propagate option, which allows the client to assign similar rights to some other entity, but ultimately it is still responsible for the incurred costs. Of course, there is no obligation for the third party to allow the delegation in the first place, but the privacy of the client is better served, if there is a possibility of delegating the authorisation to an ephemeral identifier, which is visible to the external observers of the base exchange. This kind of setting also enables the user to pay for service usage of others. Naturally, the client is also responsible for issuing a separate certificate signed with the original identity that authorises the ephemeral identifier to use the TTP certificate (see Figure 6). There should be an expiry time as well. This kind of scenario then requires that the certificates are encrypted, so that the correspondence of identifiers is not evident to the outsiders. Obviously, this does not provide anonymity towards the service provider.

```
(
 (cert
  (subject (hash sha1 <hash value>))
  (issuer (hash sha1 <hash value>))
  (validity
   (not-before 2008-07-30_08:00:00)
   (not-after 2008-07-30_09:00:00))
 )
(signature (rsa-sha1 <sig>))
)
```

Figure 6. Example of delegation certificate

The client should pay attention to the validity times and authorised amounts in the certificates, so that it has valid authorisation available, if the service requests new negotiation after the previous hash chain values have been used up to the specified maximum. At this point there is no need to do the whole base exchange again and the parties can take advantage of the HIP control packets to update the association.

3.4. Hash token handling

HIP UPDATE packets are used to transmit the next hash chain value, when it is due. This requires additions for HIP specifications. That is, a new HIP parameter needs to be defined, such as depicted in Figure 7, which identifies the used hash chain and the next value. The UPDATE must also be acknowledges with the corresponding ACK packet in order to make sure that the packet has not been lost. For added security, the parameter should be encrypted, so that someone else cannot capture the hash value and use it to pay for its own service. Naturally, the server should detect this kind of case and prevent the use as it knows which chain is related to which client, but to some less scrupulous servers just the acquisition of the token can be enough.

Type (15 bits)	C	Length (16 bits)
Hash chain id length (16 bit)		Hash value length (16 bit)
Hash chain id (variable length)		
Hash value (variable length with possible padding)		

Figure 7. HIP parameter to convey hash chains

Alternative approach would be to integrate the transmission of hash chain values into the transport protocols, e.g., IPv6 headers could be used in the use case described below. However, this would require making similar modification to every transport case for which the non-repudiation mechanism was applied. Clearly, it is easier to use more general approach with the available HIP update mechanism.

When the service wants to cash in the tokens, it contacts the third party in question and presents the given offer, the response and the relevant authorisation certificates. Also, the amount of tokens and the last received token value are submitted, so that the third party can verify the correct amount of used hash chain tokens. The third party compensates the service provider and at later point presents a bill to the client.

4. Use cases

Here we discuss potential use cases for the suggested NoRSU method. The cases presented deal with network access and streaming services.

4.1 Network attachment

The network attachment scenario is very much the basic use case for NoRSU. Thus, the idea is to "pay" one's net usage with the exchanged tokens and prove that one is authorised to receive service. This could mean, for instance, allocation of certain amount of transmitted bytes per time unit.

In the network setup we assume that we have an access point (or possibly several of them) and an access point controller, which also functions as a gateway to the external networks. The user makes the

initial attachment to the access points, but the actual base exchange is run with the access point controller. The setup resembles the architecture given in [14], which allows even the link level frames to be transmitted to the controller. Note, however, that the access points still can exhibit enough intelligence to check the validity of the puzzle solution, so that the invalid packets do not even reach the controller, hence further mitigating the denial of service concerns.

Even though we are working in the access domain and discuss mainly the interaction between the user and the access controller (corresponding the client and the server of the previous section), one could also device ways to include the home domain into the online transaction. For instance, a setup envisaged in [7] could be one alternative.

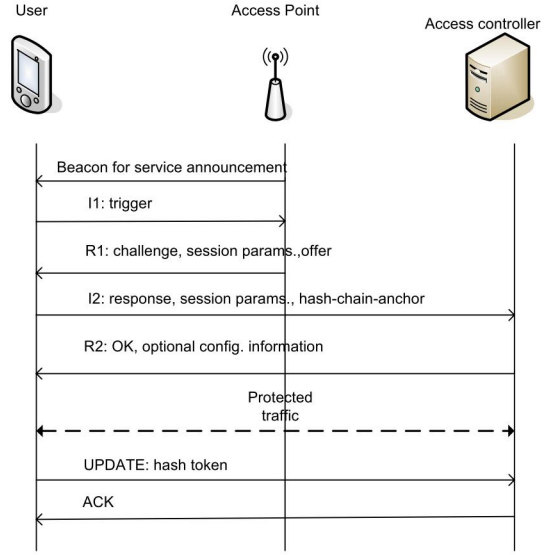


Figure 8. Attaching to network using non-repudiable service usage

The message exchange between the parties is depicted in Figure 8. We envisage the access points to be intelligent ones, so that they are able to respond to the initial I1 message with the precalculated R1 message, which also contains the offer for the network access use in the form of an SPKI certificate (see example in Figure 9). Naturally, there has been previous communication between the access point and the controller regarding the contents of R1 messages.

The user's response comes in the I2 message and it is forwarded to the controller in case the puzzle solution is correct. An example of the attached certificate is given in Figure 10. Once the controller has accepted the user as valid communication partner, R2 message is sent as an acknowledgement of the transaction as in typical base exchange. The accounting part is implemented with the help of HIP UPDATE packets as depicted earlier.

```
(
 (cert
  (issuer (hash sha1 <hash value>))
  (offer (time 1 (s 60)))
  (validity (not-after 2008-10-30_12:00:00))
  )
 (signature (rsa-sha1 <sig>))
)
```

Figure 9. Example of offer certificate

```
(
 (cert
  (hash-of-offer sha1 <hash value>)
  (hash-chain-anchor sha1 <anchor value>)
  (issuer (hash sha <hash value>)
  )
 (signature (rsa-sha1 <sig>))
)
```

Figure 10. Example of response certificate

While one might make the assumption that the network setup ensures that no traffic hijacking or redirecting can take place, it is not for certain in all environments. So, there is need to bind the actual traffic to the used identities and hash chains. The binding to the negotiated association could be done either on link or network layer, for which the base exchange provides keying material.

As discussed in [14] the link layer security can be extended all the way to the controller, which makes it transparent to the user from the network layer point of view. The similar kind of link layer setting is envisaged in the network attachment procedure described in [15] and it is based on the similar HIP alike protocol.

On the network layer the binding to the actual traffic can be done with the help of IPsec. In other words, the participants also establish IPsec association during the base exchange. As the same keying material is used to for the association setup, the binding to the tokens can be ensured. However, this basically requires that the user tunnels all the traffic to the controller that imposes extra overhead. This is similar as is envisaged to be done with Protocol for carrying Authentication for Network Access (PANA) based IPsec access control solution [16], even though key management solution is different. Using modified transport mode ESP might be one solution, but it violates the original end-to-end idea of it and requires changes to the packet processing at the controller side, i.e., it has to first process (and

remove) ESP part before forwarding the packet towards its final destination. Additionally, if the end user wishes to setup HIP associations with other hosts, one need to make sure that there is no Security Parameter Index (SPI) collisions with the existing association with the controller. Similar concerns touch Bound End-to-End Tunnel (BEET) mode [17]. Thus, tunnel mode and link layer approaches provide more feasible approach. Also, they have the possibility of protecting the privacy of the user in terms of with what other nodes it communicates. The actual binding is negotiated during the base exchange.

4.2 Accessing streaming service

Here we consider the case where a client wishes to access a streaming service provided by the server. This could be a multimedia service, such as downloading a song or a video, which is using Real-time Protocol (RTP) for executing the transport of streams [18]. Note that generally the stream is described beforehand, for example, with Session Description Protocol (SDP), but here we only concentrate on the transport part.

RTP itself provides little security, so in order to make the strong binding to the actual negotiation, we use the secure profile of RTP, i.e., Secure RTP (SRTP), which provides integrity and confidentiality services along with replay protection [19]. While using IPsec with real time traffic might be an option as well, the added latency and jitter can degrade the quality performance of such solution significantly [20]. However, many current tools might still favour IPsec due to more tried and interoperable key management.

RTP is a framework that is intended to be extensible enough to allow easy creation of profiles to meet the requirements of applications requiring transport of different kinds of real-time data, e.g., Voice over IP (VoIP). It consists of two different protocols: RTP for transporting the actual data and RTP control protocol (RTCP), which is used to report the characteristics of the connection, such as the quality of service, and convey information about the participants [18]. Hence, in very simplified terms one can consider RTP to be flowing from the server to the client and RTCP from the client to the server. Note, however, that RTP is intended to be applicable to multicast scenarios as well, although in our discussion we are concentrating on unicast transmission as HIP associations are mutual. HITs could be applied in multicast solutions, though.

There exists some work that has discussed the integration of SRTP with HIP [21], and that is the basis of this use case. Basically, the idea is to bind the RTP stream to the negotiation that has taken place within the base exchange. As SRTP leaves the question of key management open, we can use the HIP mechanisms to create the common master secret that can be used to establish the required session keys for the real-time session. [21] defines the additional parameters that have to be included in the HIP base exchange in order to achieve this, although the definitions are not yet complete. The modified protocol flow is depicted in Figure 11. Alternative is to run the offer-response interaction in the base exchange and then do the SRTP negotiation after that using the UPDATE packets. In any case, the UPDATE packets are used for re-keying.

The use of SRTP parameters provides the participants an agreement about the used encryption and authentication algorithms and their corresponding key lengths. Also, key derivation function is agreed, so that session keys can be derived from the master key and master salt. The salt is also exchanged, but the key is extracted from the keying material that is created during the base exchange (an index to the keying material can be provided). Other RTP specific parameters can be exchanged as well, such as those indicating the synchronisation source for identifying the participant and rollover and initial sequence numbers for packet indexing.

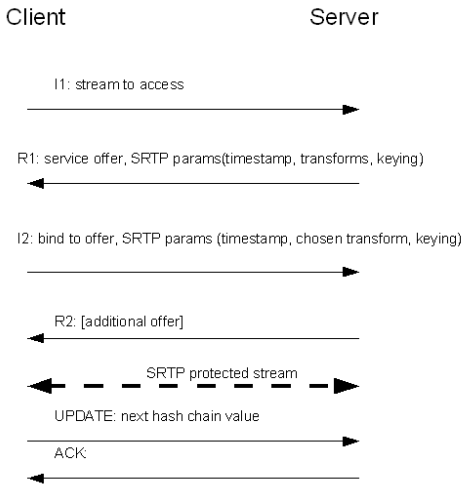


Figure 11. Using NoRSU with SRTP

However, we still need to add the non-repudiation property to this solution according to our suggested mechanisms. Thus, the I1 message already signals the client's intention to access a streaming service, so the R1 message can contain the corresponding response, which gives both the service offer and the proposed SRTP parameters. I2 contains the selected SRTP parameters along with the client's response to the offer. R2 does not contain any additional SRTP data. After

the exchange the both parties have an understanding about the frequency of the release of tokens.

During the service use the client needs to transmit the tokens to the server and this can be done using the same HIP UPDATE packets as described in the previous section. Another option that would integrate accounting more tightly with the stream itself would be to use the reporting functionality of RTCP, or more precisely, Secure RTCP (SRTCP), which provides the same services for RTCP as SRTP provides to RTP [19]. However, RTP philosophy does not take into account the acknowledgement of packets as loss of a single packet is not consider that important. Thus, detecting loss of packets transmitting the hash tokens would end up increasing the complexity. Also, as mentioned previously, we wish to prefer the general approach with the employment of HIP UPDATE.

So, in this use scenario we have described how a streaming service could take advantage of non-repudiation. The stream is strongly bound to the used identities, because the keying material used to protect the multimedia stream is derived from the negotiation done during the base exchange. That, in turn, translates to the used identities.

5. Implementation restrictions

In terms of packet size, overloading of HIP messages faces some challenges. This section talks about the relevant restrictions.

5.1 Available space in frames

While using the previously described certificates for authorisation is somewhat straightforward, one has to remember that the usage of HIP sets some restrictions. Mainly, due to the defined packet format, the length of HIP parameters is restricted to 2008 bytes, which has to accommodate the mandatory base exchange parameters, as well [2]. Also, the HIP headers (being logically IPv6 extension headers) are specified as unfragmentable, i.e., the IPv6 implementations are not allowed to fragment the header in order to meet the maximum transmission unit (MTU) of packets. This is mainly intended for avoiding DoS attacks caused by invalidly fragmented packets.

One additional limitation, i.e., around 1500 bytes, for MTU could be the use of Ethernet links on some section of the paths, but the situation can be actually worse than that. IPv6 specification states that the minimum IPv6 implementations are allowed to assume 1280-byte MTUs [22], and according to the survey done in [23] many of the current IPv6 paths seem to use it (well over 40% of the surveyed paths). Thus, length restriction due to small MTU cannot be ignored. Naturally, in environments that support higher MTUs or link layer fragmentation, like in many wireless technologies, the requirements are more relaxed, but one still needs to consider the whole path. Note that HIP is intended to be usable with IPv4 as well, where minimum MTU requirements are different, but focus of our discussion is on IPv6.

If one considers the typical HIP base exchange, it is quite obvious that the most of the information content is in R1 and I2 messages. Hence, they are the most restrictive ones for our purposes. Unfortunately, they are also the messages that will be carrying our extra payloads, i.e., offers and responses. As the amount of bytes changes from application to application, it is not easy to tell the exact amount needed for each messages, but looking at the HIP specifications and the available base exchange parameters, one can make an estimation of the used bytes. Table 1 presents mandatory HIP parameters for R1 and I2 messages along with a couple of most likely optional ones (R1_counter for indicating the current generation of the puzzle and Echo for echoing the transmitted value back), which are used to enhance the security properties of the protocol.

Table 1. Estimated sizes of R1 and I2 messages

Parameter	R1 bytes	I2 bytes
[R1_counter]	16	16
Puzzle	16	
Solution		24
Diffie-Hellman[1]	200	200
HIP_transforms	16	8
Host-id[2]	250	
Encrypted host-id[3]		280
[Echo_request/response]	20	20
HMAC		24
HIP_signature	140	140
Total	658	712

[1]Assumed just one 1536-bit D-H group (up to two groups could be proposed)
[2]1024-bit RSA key with 100 bytes domain part (which is optional)
[3]Encrypted using 128-bit AES-CBC (encryption is not mandatory)

As mentioned, table figures are just an estimation, which takes into account the "typical" parameters. The situation would be quite different, if one were to require, for instance, larger Diffie-Hellman groups and

longer keys in the name of better security (which is quite understandable, for instance, in the case of long term keys). Additionally, one might need extra parameters to signal additional associations, such as the case when negotiating the use of IPsec ESP for subsequent traffic.

As discussed in [9] the key types have significance as well. In the table above we assumed the use of RSA keys, but HIP also allows employing DSA keys. RSA and DSA keys provide roughly the same level of security for similar key lengths, but the RSA key generally has a shorter representation, because with DSA you basically also have to transmit domain parameters (this can be avoided in some special scenarios, though) [24][25]. However, the DSA signature takes less space than the corresponding RSA signature, which is dependant on the key size.

When considering these two things together, one can come to a conclusion that if one has to transmit both the key and signature, the RSA is more optimal solution length-wise. However, if it is required that only the signature is transmitted, DSA is a better alternative. Thus, this leads to a conclusion that within our context RSA is better suited for host identities, whereas the trusted third parties could use DSA keys to generate the signatures for the certificates. It is, after all, assumed that the parties have pre-established trust relationships with the trusted third party and know their relevant public keys.

The most optimal solution for this case would be elliptic curve cryptography (ECC), because it offers shorter key sizes and its performance is comparable to DSA [26]. Currently, however, HIP does not specify the possibility to use ECC keys for host identities. This should be a viable research direction for the future, especially considering the increase in key lengths over time.

5.2 Encoding

As discussed in the previous subsection, our working environment is somewhat restricted when it comes to the length of the messages. Thus, there is also need to consider the encoding of the embedded certificates. The previous examples were given using S-expressions, which, while human readable and good for examples, are unsuited for transmitting on the wire. For instance, the signatures and other binary data could be presented with base64 encoding, which clearly is wasteful when it comes to the used space.

SPKI drafts define the possibility to use canonical S-expressions, which aim at more efficient packing of the information [27]. It is also a form, which is expected to be used, when doing operations, such as hashing, on expressions. It basically presents the expressions as binary byte strings, i.e., octets, and precedes every token with the length value. This allows presenting binary data in a concise way, but still the textual tokens use the space inefficiently. In Figure 12 we present an example of canonical form of the delegation certificate given in Figure 6 (binary data omitted and line breaks added for readability). Using this encoding the length is reduced by around 60 bytes, but still the size of the certificate is around 350 bytes.

```
((4:cert(7:subject(4:hash4:sha120:<omitted>))
(6:issuer(4:hash4:sha120:<omitted>))
(8:validity(10:not-before19:2008-07-30_08:00:00)
(9:not-after19:2008-07-30_09:00:00)))
(9:signature(8:rsa-sha1128:<omitted>)))
```

Figure 12. Example of canonical encoding of S-expression (formatted for readability)

However, there is some work that considers binary encoding of SPKI certificates and that would suit our purposes as well. A SPKI authorisation certificate presented in [28] took a little over 250 bytes. It contained hashes of the issuer and subject public keys, validity dates, a simple attribute and a DSA signature, which actually could be made even more concise. In [28] it took 190 bytes, because it also contains the public key of the certifier (without domain parameters). Thus, if one makes the representation of the signature more concise, it is possible to save over 100 bytes. After all, HIP base exchange already conveys the public keys.

Table 2. Examples of records of the efficient encoding scheme

Expression type	Implied size in bytes
Subject 1024 bit rsa public key hashed using sha1	20
Subject expressed with HIT	16
Valid end date	4
Valid start and end date	8
Signature 1024 bit rsa using sha1	128
Signature 1024 bit dsa using sha1	40
Signature 2048 bit dsa using sha1	56

So, if we take into use similar kind of encoding that only has a 2-byte type field and a variable length value field per one record. The length is expected to be implicit based on the type. The type encodes much of the expression itself, e.g., we might have different

types for issuers expressed with HITs or just with SHA-1 hash values, i.e., multiple textual tokens are reduced. Also, some expressions, such as validity times can be reduced to more concise form by encoding multiple values into a record and using seconds to express dates. Thus, we are driving towards utmost efficiency at the expense of flexibility. Table 2 shows an example of some encoded expression types and the implied size of the following value field.

Table 3 shows how many bytes different certificates could take using this efficient encoding. When comparing, for instance, the efficient encoding of delegation certificate to the canonical encoding, the reduction is almost 50%. Note that encoded values could contain additional structure inside them, e.g., the encoding of the actual offer expression takes into account values such as the amount of hash tokens needed, the used unit, and the amount of units. So, one should be able to express things like 1 hash token per 60 seconds or per thousand kilobytes. A more complex offer would include the possibility to point to an external offer, which could be an XML document giving more details, but in the access scenario the client ought to be able to access the said document, i.e., have connectivity before connectivity service has been agreed on. We are, however, aiming for simplicity.

5.3 Summary of restrictions

When we consider the previous discussion regarding MTU and consumed bytes in HIP parameters, it is obvious to question, whether the suggested certificates can be included within the HIP messages. Taking into account the figures in Table 1 and amount of bytes needed for IPv6 and HIP fixed headers (both take 40 bytes), it can be concluded that with a safe margin one can use roughly 400 bytes for certificate information (R1 can contain bit more). One should also not forget the HIP parameter for signalling the non-repudiation and the target service. In a case of simple service naming (like a hash), the parameter would take 32 bytes, but with other encodings it naturally could be larger. There is room for optimisation, though. If we were to leave out the domain part of the host identity, which basically can contain Fully Qualified Domain Name (FQDN) or Network Access Identifier (NAI), we can save around 100 bytes compared to the given figures. Considering that we are planning on giving explicit authorisation for the used identities in the form of certificates, the dropping of domain part is not so crucial.

Now, if we also look at the data given in Table 3, we can come up with estimates for the amount of data added due to the certificates. If we first consider R1 message, one can expect that it contains an offer for the service, but also an authorisation issued to the server by a TTP. The table actually just gives figures for client authorisation, but the amount of needed bytes is similar. Thus, those two certificates fit within the constraints given. For I2 message one needs the response and also the authorisation ensuring the liability of the client and this should not be a problem, either. However, if we want to support the advanced scenario, where the right to use the service is delegated to another entity (or, in case of privacy protection, to another identifier of the same entity), we are hanging on the very edge of our constraints. Thus, implementations would need more care in such circumstances. The negotiation of key management procedures for additional protocols, such as those depicted in the use case of SRTP, might have to be postponed to the UPDATE messages after the base exchange has completed.

If such additional roundtrips are undesirable, there is still room for further optimisation in the used certificates. As the HIP packets already contain signatures of the client and the server, the end point generated certificates for offers and responses can do without signatures. This saves well over 100 bytes (in case of RSA signatures) and enables one to fit all the authorisation statements within the limits we have set. The downsides of this approach are increased storage requirements and added complexity, because the parties need to store the whole HIP packets instead of a set of certificates. The clearing party has to also be able to understand HIP structures.

It is worth noting that the previous discussion assumes 1024-bit keys, whereas longer keys make it even harder to fit the information within the HIP

Table 3. Byte count for different certificates

Offer	**bytes**	**Response**	**bytes**	**TTP-client**	**bytes**	**Client deleg.**	**bytes**
Issuer	22	Hash-of-offer	22	Subject	22	Subject	22
Offer	6	Chain-anchor	22	Issuer	22	Issuer	22
Validity-end	6	Issuer	22	Target serv.	22	Validity-range	10
Rsa-sign-1024	130	Rsa-sig-1024	130	Amount-max	6	Rsa-sig-1024	130
				Propagete	2		
				Validity-range	10		
				Dsa-sig-2048	56		
Total	164		196		140		184

header. It should be noticed, though, that the constant increase in computing power and the developments in the mathematical algorithms is bound to raise the bar for the required key lengths. The 1024-bit keys are considered to be adequate for the next couple of years, but scenarios needing longer term solutions, such as those related to the trusted third parties, should already use 2048-bit keys [29]. This further motivates the need to look into the possibility of using ECC keys.

One additional length consideration relates to the use of hashes. Even though SHA-1 is a very common hash function, it is showing some weaknesses [30]. Therefore, one should also consider the use of advanced forms, such as SHA-256, instead in places that require hashing of relatively free form messages. The increase in length is just 12 bytes, though, but can build up when used in multiple places Note, though, that when hashing public keys and the attacker wants to find another key that hashes to the same value, it is very unlikely to find such a value, because one is not able to modify the source value at will and still retain the required structure for a public key. This also applies to the case of HITs, even though they are shorter than SHA-1 hash values. The case where this matters most is the hashing of the offer of the server to indicate to which offer the client is binding itself.

6. Analysis

The following subsections analyse the potential threats and the corresponding countermeasures from the viewpoint of our proposal.

6.1 Threats within the context of the solution

In this section we discuss the potential threats that can emerge in an environment that plans on adopting the suggested token based solution. One should note that not all of them have a technical countermeasure, but those should then resort to other measures offered by the society in case of agreement dispute, such as litigation. While this subsection concentrates just on listing the threats, the way the countermeasures take place is discussed in the next one.

As has been described earlier the interaction is mainly between the user and the service, but from the threat analysis perspective one has to also remember the existence of a third party, such as the home operator, who acts as a trust and liability broker between the entities. There might also be an external entity, who could try to interfere with the service provisioning.

In the case of compensation of the service usage, the setup can pose several threats to different parties. The most obvious threats are that the service is not paid for or that the service is not received after paying. Especially when the user pays after the service usage (post-paid), there is a chance that the user repudiates it, i.e., claims that he has never used the service and the cost claims are unfounded.

Different collusion scenarios can be envisaged. In other words, two of the parties conspire against the remaining one. The home operator could assure the trustworthiness of the user without any intention of compensating the service afterwards at the time of the clearing. On the other hand, the user and the service could collude against the home operator in order to make the home operator compensate the service without the user having no intention of paying his bill later on. While being perhaps the most unlikely case of these, the service and the home operator could try to make the user pay more than the user originally thought (misleading advertising is another matter).

Double spending can occur when otherwise valid tokens are replicated to pay for several different transactions. In a similar sense, the service might try to charge the accessed service more than once. Very close is also the case of overspending, when the user consumes more resources than he can afford, i.e., overly large amount of tokens is created and used.

Hijacking of information by an unauthorised party could also take place. The payment tokens or the actual payment could be stolen by some other service or another user could try to use the tokens to pay for his service. Also, instead of tokens, another user could try to hijack the paid service from the legitimate user.

Integrity of the compensation agreement could be facing threats, as well. Either the user or the service provider might try to modify the agreed terms, so that the later claims would be more favourable to them. This also includes forging of additional tokens so that the service could claim more resource usage than really took place.

User privacy is always an existing threat in any communication system and it will become even more important as the transition towards ubiquitous communication takes place. This is especially evident in our proposed solution, which makes heavy use of different kind of identities. User privacy is at stake if the identity information is disclosed to unauthorised parties, who are then able to track the users, for instance. Also, users may also wish to prevent others from learning what sort of services they are using and what sort of usage patterns they follow. Thus, the users should be in control of the disclosure of information about themselves and their actions.

As mentioned above, within the limits of our proposed solution, some of these threats can only be addressed through litigation. For instance, if some party refuses to pay even when faced with technical evidence, the other parties have to initiate legal procedures in order to get the promised compensation. This is not, however, different from the case, where a user refuses to pay his post-paid subscription or credit card bill. This is a business risk, which should be embedded in the business models of the players.

Generally, when faced with collusion of other parties, the legal action with the technical evidence is the only solution. Naturally, fear of losing one's reputation can be enough disincentive for the home operator to not to cheat the user as the user trust is the very foundation of its business model. In the following section we analyse the properties of our proposal, which can provide solutions to the technical threats presented above.

6.2 Technical measures against the threats

The basic components in the proposed solution are the use of hash chains and the binding of the identities to them. The hash chains provide the means to pay for the service usage in a piecemeal fashion, i.e., as long as the service is received, additional hash chain values can be submitted. Analogously, as long as the server keeps receiving new hash values that are part of the chain, the service is provided. Thus, in case of malicious party, no further compensation or resource provisioning is provided, i.e., the granularity of the commitment is better controlled. Generally, the value of a single hash token should be kept small as that is the amount that can be lost in the case of misbehaviour.

The hash chain values have the added benefit of being easily verifiable, because the receiver has to only compute one hash function in order to make sure that the received value is part of the chain. Naturally, the used hash function has to be secure enough, so that the receiver is not able to calculate future values. This ensures that only the entity that has knows the secret seed of the hash chain knows the transmitted values beforehand. Hence, the service provider cannot easily create additional tokens, so that it could make cost claims for unused resources. Also, as the value of single token is kept small, the required effort of brute force attack clearly outweighs the benefit.

Non-repudiation property of the solution comes from the binding of the identities to the presented offers and responses. When the user presents the anchor value of the hash chain, he has signed the statement with his identity and also included in the statement the reference to the received offer. This, along with the assumption that the hash function is irreversible, dictates that only that user has been able to create the said hash values. Thus, if the user denies using the service, the service provider only needs to present the anchor value signed by the user and the last received chain value in order to prove that the user has used service with the offered terms. The service provider can naturally deny that it has provided any service, but from the point of view of our solution concept it does not matter as the user already has consumed the desired resources. The service, however, cannot deny that is has given a service offer on certain terms.

The trust to the client's ability to pay comes from the associated TTP certificate, which authorises the client to use a certain maximum amount of commodity. This can be seen as the credit the client has in the eyes of TTP and as an acknowledgement that TTP knows the client. This way the server has certainty that someone will provide compensation for the provided resources, because ultimately TTP has accepted the liability in the case of misuse when issuing the certificate to the client. It is then up to the agreement made between the third party and the client to settle the costs. This does not differ from the typical post-paid business model commonly used in the telecom or credit card industry.

It is also possible to grant an authorisation certificate for the service as well, to be presented as proof of its trustworthiness. Although, as stated above, in case the server is not providing the service it promised to deliver, the client just can stop sending any hash chain values. If the service in question is other than the typical access scenario, like buying a song, then the motivation to include such authorisation might be different. This is basically a risk management decision for the client. Of course, if the song, for example, is streamed, then the client has better control of what it is receiving and can pay it piecemeal, like in the second use case scenario described earlier.

The employment of HIP provides a natural way of taking advantage of the accompanying cryptographical identifiers for presenting the identities of the parties. As it also provides a key management solution, it can be used to create the necessary association so that the actual data traffic can be bound to the same identities, even though it requires an existing data traffic protection mechanism, such as IPsec. Thus, even though IP addresses could be spoofed, the mutually agreed keying material ensures that the traffic is useful only to the valid partners.

Fine tuning of the solution is done through the extra attributes given in the certificates and the procedures the parties conduct during the transaction. When the client clearly states the service provider identity in the response message, no other service provider can claim the costs, even though it somehow could manage to get hold of the hash tokens. It is possible for the home operator to give more granular authorisations to the client and only allow certain service providers or service types. One should remember, though, that if the same authorisation allows the use of several service providers for the specific service type, the maximum allowed resource consumption could not be controlled without online access to the home operator, which tends to complicate things and decrease performance. However, this can be found out during the clearing procedure and extra claims made towards the user. This is basically a risk management decision for the home operator, when deciding what sort of granularity to use in the issued authorisations.

In any case, the server has to remember to check the provided anchor values, so that it is not possible for the client to use the same anchor value within the validity period of the same offer. Otherwise the client might be able to use the same hash chain values again, but the server would only be able to bill them once. This could also be prevented by having an individual session identifier in every R1, but cannot be done without breaking the basic HIP properties as the signature in R1 is pre-computed for the sake of mitigating denial of service possibility.

The privacy of the client is preserved through the decoupling of authentication and authorisation. TTP issued certificate can state that the ephemeral identifier assigned to the client is trustworthy for certain actions. Naturally, TTP is able to connect this to a real identity. Also, this introduces additional overhead in terms of additional interaction with TTP, especially if the

Table 4. Threats and possible countermeasures

Technical threats	Preventive measures	Threats handled through litigation
User denies having used the service	Binding of identity to the hash chain	• User and home operator collude against service • Server and home operator collude against user • User and service collude against home operator • User is charged too much at the time of clearing • User refuses to pay at the time of clearing • Service does not get money from the home operator at the time of clearing • Privacy of the user (e.g., home operator releases information about the user without user consent)
Service provider does not provide agreed service	Stop transmitting additional hash tokens	
User gets no or other service that he paid for	Stop transmitting additional hash tokens	
Service hijacked by another user	Bind the payloads to the negotiated keying material	
Intercepted tokens used by another user to pay for his service	Service needs to ensure the strong binding between the identity and the hash chain, protection of tokens with the negotiated association	
User double spends the created compensation tokens	Authorise specific service, check anchor value uniqueness	
Service charges user multiple times	Non-repudiable accounting records are accepted only once	
Other service "cashes" the tokens	User authorises specific provider	
Service creates additional valid user tokens	Secure hash function prevents creating additional usage records and user signature protects the hash anchor value	
User modifies the offer to more favourable one	Offer is protected with signature	
Service modifies the offer to more favourable one	Offer is protected with signature	
User overspends his credit	User is authorised only to spend certain maximum amount	
Privacy of the user	Use of ephemeral identities and delegation	

identifier is changed often. This allows providing anonymity towards the service providers, though. The other option is that the TTP provides an authorisation for the long term identifier and the client constructs an ephemeral identifier for which it delegates the authorisation. Even though this is a more flexible option, it does not provide complete anonymity towards the service providers, because they need both the delegation and the original authorisation. However, it does, like the other alternative, provide privacy protection against external observers, because the real identity does not have to be visible, not even in the form of its HIT.

In Table 4 we have summarised the different kinds of threats that might emerge in this kind of service usage concept. Additionally, the table present how the suggested solution can answer to the technical threats.

7. Conclusion

In this paper we have proposed a simple accounting scheme to be used in conjunction with HIP. With his kind of solution the service provider is able to get undeniable evidence that it is entitled to compensation for the provision of its resources to a certain client. On the other hands, the client can control the charging procedure, so that it is only billed for the costs that are based on the actual usage.

We have showed that the combination of HIP and hash chains can provide a secure solution for non-repudiable service usage that also takes into account the binding to the actual data traffic. This is further enhanced with the employment of authorisation certificates to increase the level of trust the client and server have for each others. Thus, it also provides an authorisation mechanism for the participants.

However, the used environment poses some problems, namely in the form of length restrictions, that need to be taken into consideration. We have considered the most common IPv6 path MTU, 1280 bytes, and concluded that even though it is possible to introduce the solution to this environment, the advanced scenarios providing better privacy support and the delegation of service authorisation may face difficulties with certain implementations. There is a possibility for length optimisation, although at the expense of increased complexity. Also, the choice of used key types has considerable impact on that and RSA based host identities are better suited for the most length restricted cases. This still calls for efficient encoding mechanisms, which have the downside of limiting the flexibility.

It is worth remembering, however, that the wide adoption of HIP is still years away, so the restrictions set by the current MTUs can be quite different in the future networks. It shows, though, that this is additional incentive for pushing for higher MTUs. Also, the research done with technologies that provide shorter key lengths, such as ECC, provides measures to answer to these restrictions, even though the requirement for having larger key sizes goes hand in hand with the increase in computing power. In any case, the host identity enabled environment provides many interesting directions for the development of secure charging schemes.

Acknowledgment

The author wishes to thank Tuure Vartiainen and prof. Jarmo Harju for comments and suggestions.

References

[1] Federal Communication Commission, "Unauthorized, Misleading, or Deceptive Charges Placed on Your Telephone Bill - Cramming", FCC Consumer Facts, online article, available in http://www.fcc.gov/cgb/consumerfacts/cramming.html (accessed 01/2009), Jul 2008.

[2] Moskowitz R., Nikander P., Jokela P. (Ed.), Henderson T., "Host Identity Protocol", IETF RFC 5201, Apr 2008.

[3] Moskowitz R., Nikander P., Jokela P., "Using ESP transport format with HIP", IETF RFC 5202, Apr 2008.

[4] Heer T., Varjonen T., "HIP Certificates", IETF Internet-Draft draft-varjonen-hip-cert-01 (work in progress), Jul 2008.

[5] Laganier J., Vicat-Blanc Primet P., "HIPernet: A Decentralized Security Infrastructure for Large Scale Grid Environments", The 6th IEEE/ACM International Workshop on Grid Computing, Nov 2005.

[6] Tschofenig H., Nagarajan A., Ylitalo J., Shanmugam M., "Traversal of HIP aware NATs and Firewalls", IETF Internet-Draft draft-tschofenig-hiprg-hip-natfw-traversal-02, expired, Jul 2005.

[7] Heikkinen S., Priestley M., Arkko J., Eronen P.,Tschofenig H., "Securing Network Attachment and Compensation", Proceedings of the Wireless World Research Forum Meeting (WWRF#15), Nov 2005.

[8] Blaze M. et al., "TAPI: Transactions for Access Public Infrastructure", Proceedings of Personal Wireless Communications (PWC2003), Sep 2003.

[9] Heikkinen S., "Non-repudiable service usage with host identities", Proceedings of the Second International Conference on Internet Monitoring and Protection (ICIMP07), Jul 2007.

[10] Lamport L., "Password authentication with insecure communication", Communications of the ACM, vol. 24, no. 11, 1981.

[11] Tewari H., O'Mahon D., "Multiparty micropayments for Ad Hoc Networks", Proceedings of the IEEE Wireless Communications and Networking Conference, Mar 2003.

[12] Zhou J., Lam. K., "Undeniable Billing in Mobile Communication", Proceedings of 4th ACM/IEEE International Conference on Mobile Computing and Networking, Oct 1998.

[13] Nikander P., Laganier J., Dupont F., "An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID)", IETF RFC 4843, Apr 2007.

[14] Calhoun P. et al., "Light Weight Access Point Protocol", IETF Internet-Draft draft-ohara-capwap-lwapp-04 (work in progress), Mar 2007.

[15] Rinta-aho T. et al., "Ambient Network Attachment", Proceedings of 16th IST Mobile and Wireless Communications Summit, Jul 2007.

[16] Parthasarathy M., "PANA Enabling IPsec based Access Control", IETF Internet-Draft draft-ietf-pana-ipsec-07 (work in progress), Jul 2005.

[17] Nikander P., Melen J., "A Bound End-to-End Tunnel (BEET) mode for ESP", IETF Internet-Draft draft-nikander-esp-beet-mode-09 (work in progress), Aug 2008.

[18] Schulzrinne H., Casner S., Frederick R., Jacobson V., "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 3550, Jul 2003.

[19] Baugher M., McGrew D., Naslund M., Carrara E., Norrman K., "The Secure Real-time Transport Protocol (SRTP)", IETF RFC 3711, Mar 2004.

[20] Bou Diab W., Tohme S., Bassil C., "Critical vpn security analysis and new approach for securing voip communications over vpn networks", Proceedings of the 3rd ACM workshop on Wireless multimedia networking and performance modeling, Oct 2007.

[21] Tschofenig H., Shanmugam M., Muenz F., "Using SRTP transport format with HIP", IETF Internet-Draft draft-tschofenig-hiprg-hip-srtp-02 (expired), Oct 2006.

[22] Deering S., Hinder R., "Internet Protocol, Version 6 (IPv6) Specification", IETF RFC 2460, Dec 1998.

[23] Wang Y., Ye S., Li X., "Understanding Current IPv6 Performance: A Measurement Study", Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC'05), Jun 2005.

[24] Eastlake D., "RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS)", IETF RFC 3110, May 2001.

[25] Eastlake D., "DSA KEYs and SIGs in the Domain Name System (DNS)", IETF RFC 2536, Mar 1999.

[26] Cronin E., Jamin S., Malkin T., McDaniel P., "On the Performance, Feasibility, and Use of Forward-Secure Signatures", Proceedings of the 10th ACM conference on Computer and communications security, Oct 2003.

[27] Ellison C. (Ed.), "Simple Public Key Certificate", IETF Internet-Draft draft-ietf-spki-cert-structure-06.txt, expired, Jul 1999.

[28] Arbaugh W., Keromytis A., Farber D., Smith J., "Automated Recovery in a Secure Bootstrap Process", Internet Society 1998 Symposium on Network and Distributed System Security, Mar 1998.

[29] National Institute of Standards and Technology, "Recommendation for Key Managent - Part 1: General (Revise)", NIST Special Publication 800-57, May 2006.

[30] Wang X., Yin Y., Yu H., "Finding Collisions in the Full SHA-1", Proceedings of Crypto'05, Aug 2005.

Publication P4

Heikkinen S., Siltala S., ”Service Usage Accounting”, in *IEEE Vehicular Technology Magazine*, vol. 6, iss. 1, pp. 60-67, 2011.

Seppo Heikkinen and Santeri Siltala

SERVICE USAGE ACCOUNTING

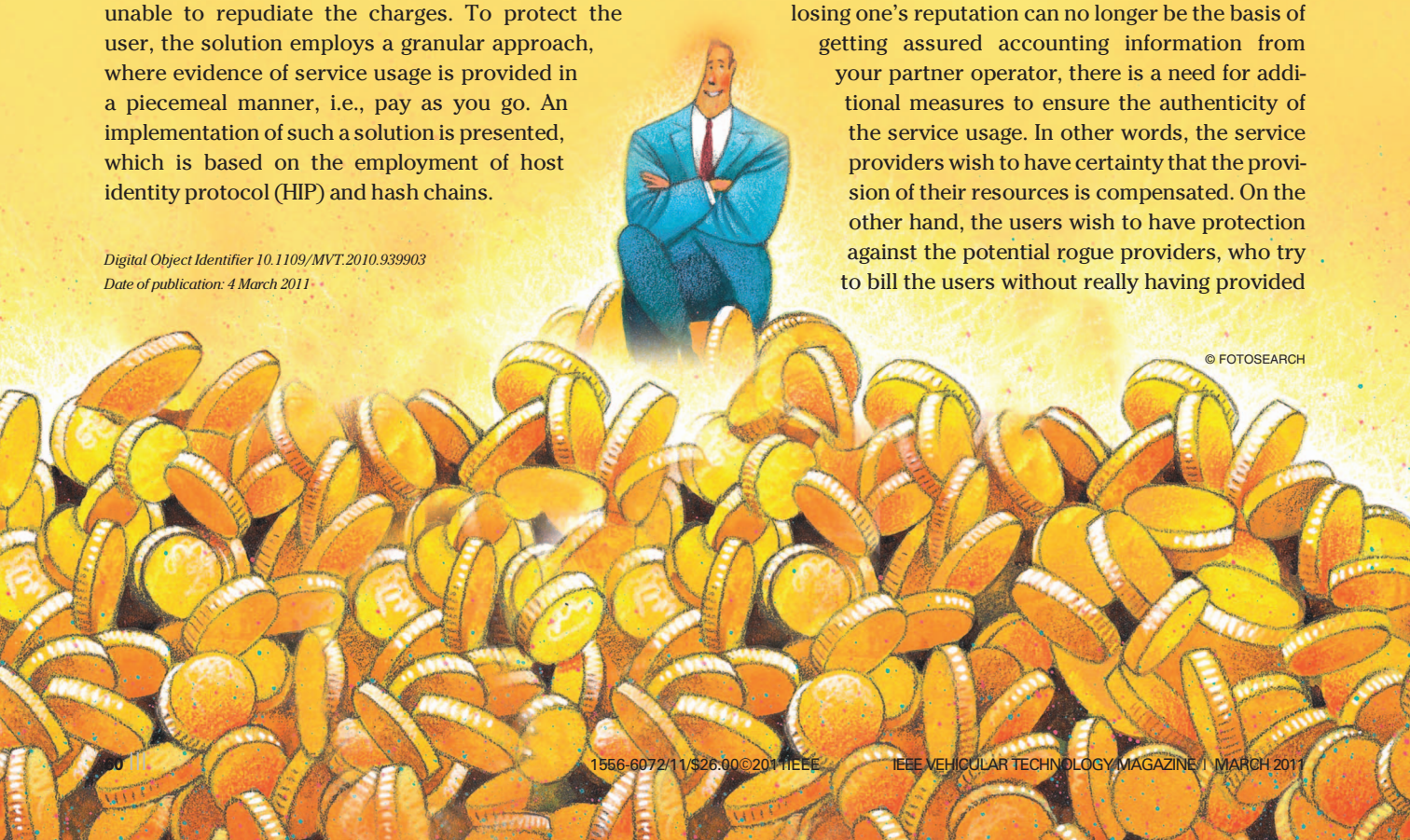
An Assured Solution for Future Networks

When a party provides service and wishes to receive compensation for the provision of its resources, the question of getting assured accounting information also emerges. Although various prepaid or postpaid solutions can be devised, it does not provide very good protection if the service is not received after the payment or the user disputes the bill. We present, in this article, a network-level, service-usage solution, which provides assured accounting information strongly bound to the host identity, so that the user is unable to repudiate the charges. To protect the user, the solution employs a granular approach, where evidence of service usage is provided in a piecemeal manner, i.e., pay as you go. An implementation of such a solution is presented, which is based on the employment of host identity protocol (HIP) and hash chains.

Digital Object Identifier 10.1109/MVT.2010.939903
Date of publication: 4 March 2011

With the advent of ambient and ubiquitous computing, there is a trend toward a more dynamic networking environment within the wireless domain as opposed to the current strict and static relationships instilled by the incumbent operators. The relative freedom of choosing your own service provider, already long established in the Internet domain, is also entering this market, when network convergence proceeds further, and the smaller players also have the opportunity to act as operators.

However, when the static relationships and the fear of losing one's reputation can no longer be the basis of getting assured accounting information from your partner operator, there is a need for additional measures to ensure the authenticity of the service usage. In other words, the service providers wish to have certainty that the provision of their resources is compensated. On the other hand, the users wish to have protection against the potential rogue providers, who try to bill the users without really having provided



1556-6072/11/$26.00©2011IEEE

any service. After all, cramming, i.e., unauthorized charges on a customer phone or credit card bill are not unheard of.

In this article, we present an authorization solution that takes into account the identities of the parties involved in the service transaction and also provides assured accounting information, so that a party cannot deny, i.e., nonrepudiate, its involvement in the transaction. The solution could be employed, for instance, in a simple network attachment case, such as access to a hotspot, or in a case involving consumption of a specific service, such as streaming of media content. Accounting is provided in a granular way, to embrace the pay as you go kind of setting. As building blocks, we employ HIP and hash chains to implement our proposed system, and use these mechanisms, along with simple public key infrastructure (SPKI) certificates, to bind various parts of the interaction, including the data traffic itself, in a cryptographically secure way. Additionally, remote authentication dial-in user service (RADIUS) mechanisms are integrated to provide the means to convey the accounting evidence to the third parties and get online consent for the service transaction.

Fundamental Technologies

HIP

HIP is a proposal for future network architectures, and it introduces a new namespace for the current Internet architecture [1]. This takes place with a conceptual identity layer that resides between the network and transport layers. This enables one to decouple the dual role of IP addresses, i.e., their functionality both as end-point identifiers and locators. Thus, better mobility and multihoming solutions can be devised. In HIP, the host identity is manifested with host identity tag (HIT), which is a representation of the public key owned by that entity, and it can be given a nonroutable IPv6 interpretation. In essence, it is a hashed representation of the public key, hence, giving HIT self-certifying properties.

Identity exchange and association between the communicating parties is established through a handshake procedure, base exchange (BEX), which also acts as a simple Diffie—Hellman (D-H)-based key-exchange solution. It provides keying material for protecting the subsequent communication with, e.g., IP security protocol (IPsec). It also takes measures to provide denial of service (DoS) resistance with client puzzles.

Hash Chains

Hash chains have not only been used in several micropayment schemes to provide payment and nonrepudiation (see, e.g., [2]) but also have various other authentication uses. Originally, they were presented in 1981 by Lamport for one-time password authentication [3].

The idea behind hash chains is based on the irreversible nature of hash functions. First, you create a secret seed value and apply the function successively. The final value is the anchor of the chain, and the previous chain values can be released in the reverse order. Thus, it is easy to check whether the received value is part of the chain but difficult, provided the hash function is secure, to calculate the next value of the chain without the knowledge of the initial seed. The check can also be done even if some in-between values are missing.

WITH THE ADVENT OF AMBIENT AND UBIQUITOUS COMPUTING, THERE IS A TREND TOWARD A MORE DYNAMIC NETWORKING ENVIRONMENT WITHIN THE WIRELESS DOMAIN AS OPPOSED TO THE CURRENT STRICT AND STATIC RELATIONSHIPS INSTILLED BY THE INCUMBENT OPERATORS.

SPKI Certificates

SPKI is intended to provide a simpler alternative to X.509 [18] certificates, even though they have not gained wide popularity. This is partly due their more flexible structure, which is not as strictly specified as when using ASN.1 [19] with X.509. However, this makes them well suited for prototyping in a research environment, where one might want to convey authorization statements without heavy ASN.1 processing. In fact, the purpose of binding a key holder to a specific authorization has been one of the main points of SPKI certificates [4]. One might claim, though, that their interoperability suffers from the lack of precise definitions and transport structure considerations, as these specifications were never finalized [5].

Nonrepudiation

Purpose of Nonrepudiation

Certain transactions have the requirement that the parties should not be able to deny their involvement in the said transaction. This could relate, for instance, to electronic commerce, where there is a clear incentive to make sure that the one party has paid and the other party has delivered the goods. A contract signing is also a traditional example. The participants prove with their signatures that they have seen the contract and approved it.

The following characteristic can be required of a nonrepudiable message transfer [6]:

- nonrepudiation of origin (NRO)
- nonrepudiation of receipt (NRR)
- nonrepudiation of submission (NRS)
- nonrepudiation of delivery (NRD).

NRO is intended to provide the proof that the sender has sent the message; thus, it cannot later deny sending it. Similarly, NRR prevents the receiver from denying

the reception of the message. NRS and NRD relate to the actions of the delivery agent and the proofs that it is working as expected. Our solution concentrates on providing NRO and NRR, even though we do not consider the evidence of service provisioning per se but expect the reception of correct service to be sufficient for the user (such additional evidence could be easily included in the corresponding acknowledgement messages, though).

Fairness

Various ways of implementing nonrepudiation system can be devised, but it is another question whether the system is fair to all the parties. In other words, it should not discriminate against a correctly behaving party [7]. A simple exchange of signed messages is not fair in the sense that once the other party receives the signed message that party can terminate the protocol, thus leaving the initiator without any evidence. To overcome these shortcomings, quite often a third party is used online to ensure the fairness of the transaction.

Rather than devising complex protocol exchange, we employ granularity in our approach to fairness, i.e., the service is paid one small piece at a time. While perhaps not meeting the formal fairness criteria, it provides a practical approach for the parties to be in better control of the involved risk without necessary online involvement of third parties.

Evidence and Accounting

Evidence is the central piece of nonrepudiation for proving the participation in the transaction and for ensuring fairness. In fact, it has been stated that the goal of nonrepudiation is to collect, maintain, make available, and validate irrefutable evidence [8]. This can also be seen from the different stages in the process of nonrepudiation [8]:

- evidence generation
- evidence transfer, storage, and retrieval
- evidence verification
- dispute resolution or arbitration.

As evidence needs to be transferred between different parties and stored accordingly, there is a need for accounting mechanisms. Typical accounting protocols are RADIUS and Diameter, although simple network management protocol (SNMP) and common open policy service (COPS) can be employed as well, at least to a certain degree [9]. However, they do not consider the strong binding of evidence to the subjects per se but instead concentrate on the message transfer. We have integrated RADIUS into our solution so that inherent security properties of HIP can be used to protect the RADIUS exchange as well, without having to worry about the manual configuration of RADIUS-shared secrets. The choice of RADIUS was based on the availability of tool kits and its widespread use, for instance, in wireless local area network (WLAN) service provisioning. This approach also allows us to later engage in roaming scenario experiments with local community networks employing RADIUS infrastructure, such as Wireless Tampere.

Nonrepudiation Protocols

A classical example of fair nonrepudiation protocol is given by [6], and it uses a third party at the end of protocol run to ensure that both the parties are able to get their allotted evidence. Numerous other similar proposals exist, but quite often, they do not consider how these would work in a real networking environment. This can reveal unexpected impracticalities in the protocol design [10]. For instance, in the aforementioned case, it is not so evident how long the evidence should be available for download from the third party. One should also note the relation between the evidence and the nature of data: it is clearly infeasible to provide and store separate evidence for every single packet. This is where hash chains become useful.

Reference [11] is an example of a proposal, which uses hash chains in global system for mobile communications (GSM) environment to provide an undeniable billing solution for calls. The actual coupling with GSM signaling protocols is not made clear, though. Home operator endorsement for used hash chains is needed in [12], which additionally requires smart cards to ensure the security of micropayments for ad hoc network routing. Hash chains as micropayment solution for WLAN access is considered in [13] and it also makes the observation of the importance of risk management. Unlike the previous ones, which rely on signatures, [14] suggests using a shared secret with a trusted party to bind the user to the nonrepudiation process in GSM setting. Obviously, with shared secret approaches, a degree of doubt of the origin exists, and the adjudicator of the evidence needs to be able to trust the trusted party as well.

Architectural Overview

We take advantage of the aforementioned HIP BEX, which forms the backbone of our identity-based approach and provides the negotiation step, during which the keys for protecting the subsequent service traffic and bindings to the nonrepudiable evidence are established. As evidence tokens, we employ the values of a hash chain.

Thus, with our enhanced BEX we have devised a system, which provides the following characteristics:

- identification of the parties securely
- liability relationships between the parties
- offline/online authorization from the home network
- negotiation of the service terms
- nonrepudiable evidence
- granularity of the service usage
- reporting the usage.

We assume as a prerequisite that the client and the trusted third party (TTP) have had previous contact and that the client has been assigned a certificate that states

its subscription relationship. Hence, TTP could be seen as a home operator. In addition, we assume that the server and TTP are able to trust each other enough to agree on business transaction. This can take place directly or with the help of some additional, common TTPs such as financial brokers. Thus, our approach ensures that the liability aspect is taken into account. This means that it is possible to find a party that is responsible for the incurred costs, even though a party, like the user, misbehaved. In a sense, this is analogous to the traditional credit-card scenario, although with online shopping and fraud possibilities, it is nowadays a bit more convoluted field.

Nonrepudiable Service Usage

Nonrepudiable service usage (NoRSU) takes place with the addition of some extra parameters to the typical HIP BEX messages, mostly in the form of application certificates. These certificates convey the relevant authorization information and can be later used as part of the evidence to the charging process. BEX with NoRSU enhancements is depicted in Figure 1. Note that we have named our entities client and server, instead of initiator and responder as in the HIP vocabulary. NoRSU-specific data are shown in bold and the HIP message types in italics.

The first message, I1, signals the willingness of the client to initiate the corresponding service usage with the help of nonrepudiation properties. As a response, the server provides its service offer in R1 in the form of an SPKI certificate, which dictates the frequency of the hash tokens it expects to receive as evidence. The frequency can relate to either time- or volume-based accounting. This message can also contain an authorization certificate issued by TTP stating that the service in question is a legitimate one. However, this is not entirely necessary, because for the client the fact that it receives the service can be sufficient, although one could claim that this allows reselling of resources.

The I2 message binds the client to the offer with the help of a signature, i.e., the client agrees to the given offer. The client also provides a hash anchor for the hash chain it intends to use during this service transaction. In addition, the client attaches an authorization certificate issued to it by TTP so that the server can get offline knowledge about the liability of the client. This can be further enhanced with the online RADIUS procedures described in the following subsection. Some additional enhancements are also possible. This message could also contain a delegation certificate, with which one could provide privacy protection or gifting, i.e., identity in the delegation certificate pay for the service usage of the externally visible identity.

R2 basically just concludes the exchange, but some additional scenarios are possible. At this point, the identity and the business relationships of the client are known; thus, additional offers could be made to valued customers. However, this would require additional signaling. This

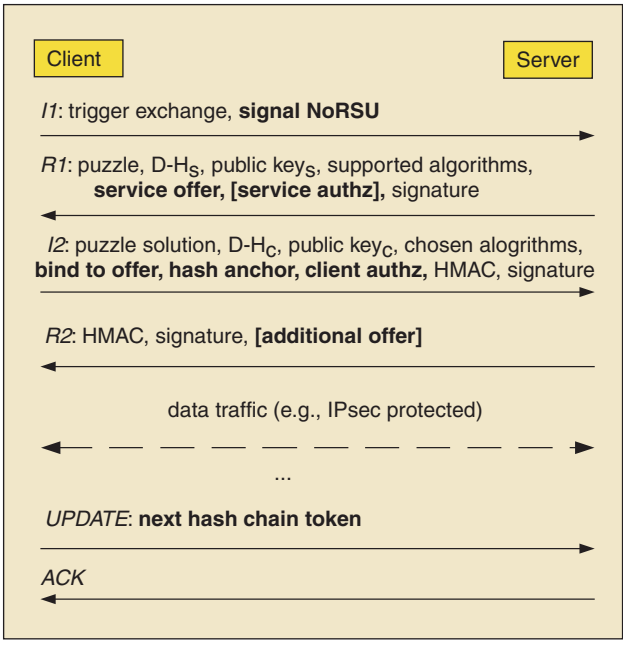


FIGURE 1 HIP BEX with NoRSU enhancements.

message could also be used, in conjunction with RADIUS exchange, to convey a challenge issued by the home operator. If one also wanted to provide proof that the server has provided service, the server could create its own hash chain and send the anchor value here and the acknowledgements to the HIP UPDATE messages would then carry these hash values. However, we did not consider this scenario any further and it will be a topic for future work.

Once the negotiation is done, the service traffic can flow between the parties, protected with the keys negotiated during BEX. The client uses HIP UPDATE messages to transfer the next hash chain token at agreed intervals, and the server acknowledges this with the corresponding message.

RADIUS Enhancements

Figure 2 depicts the integration of RADIUS into our solution. The figure is used only to sketch the potential messages that could be exchanged with the involved parties and in case TTP was not HIP capable, the server could just use normal RADIUS messaging. However, security for RADIUS should be then offered in some other way. Note that, instead of our somewhat straightforward setting, the entity interacting with TTP could be a middle box, which does not serve as a service end point for the client but just observes the traffic and makes its own decisions about allowing or blocking the traffic (e.g., firewall functionality).

Initially, the interaction proceeds as in normal NoRSU exchange, i.e., the offer is given in $R1_A$ and the corresponding response in $I2_A$. After that, the server has the option of requesting extra confirmation from the party who had issued the TTP certificate for the client. Using the issuer information, the server discovers the means to contact

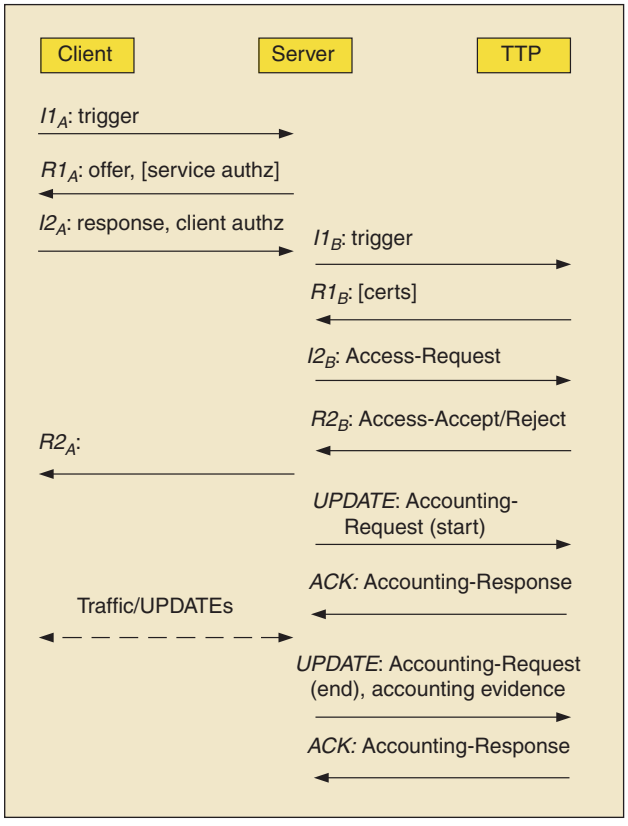


FIGURE 2 NoRSU complemented with RADIUS support.

TTP. This could take place with the help of domain name system (DNS), distributed hash table (DHT), or by using the information in the certificate provided by the client.

If the server does not have an association with the TTP, it will establish it using the typical HIP BEX. The association could have been formed previously when dealing with another client of the same TTP or when having another transaction with the same client. In this case, HIP UPDATE packets are used (even though the IPsec-protected RADIUS traffic could also be possible). In the former dynamic case, TTP may present some additional certificates if its identity is unknown to the service or stronger liability guarantees are needed. $I2_B$ and $R2_B$ are then slightly modified to carry encapsulated RADIUS messages, which convey the request for extra confirmation. This could be needed, for instance, to alleviate any revocation concerns and get the online consent from TTP about the current status of the client. This could be used when TTP does not provide the TTP certificate to the client, or the TTP could specifically instruct the server to use online procedure in the certificate itself. All the RADIUS messages within the HIP parameters should be encrypted.

$I2_B$ contains the access-request message, with which the server identifies the client with a HIT and queries whether the client in question is authorized to use the service. However, if we want to prevent the server from freely enquiring the customer relationships of TTP, we have to include the response certificate as evidence that the client has been present, even though the granularity of this proof of freshness is rather coarse (the validity time of the response is set by the client, within the limits of the given offer).

In $R2_B$, TTP replies either with access-accept or access-reject message to either allow or deny the service provisioning. In an advanced scenario, TTP might issue a challenge to verify the identity of the client, especially if it is not content in receiving just the response certificate as proof of freshness. In such cases, one would communicate the challenge between the server and client and use $R2_A$ to convey this information. Subsequent UPDATE messages would have to be used to finalize this exchange and get the final accept or reject. $R2_A$ could also convey a signed message from TTP to the client, stating that TTP approves the identity of the server as a suitable transaction partner (this would naturally be present in $R2_B$ as well).

After submitting $R2_A$, the server can start the accounting with TTP by issuing accounting-request within an HIP UPDATE message. Corresponding ACK contains the accounting-response message. After the traffic exchange between the client and server has terminated, accounting-request is sent to TTP along with the information that tells how many hash chains were consumed along with the other evidence information.

The other evidence consists of the offer and the response certificates but could also contain a delegation certificate if such was used. In addition, one needs to convey the values that were used to indicate the renewal of hash chain in case the previous hash chain was exhausted.

In essence, we need at least the following data as evidence:

- offer certificate
- response certificate
- client authorization certificate
- amount of hash chain values transmitted
- first and last hash chain value transmitted.

Note that it is easy to confirm the amount of hash chain values expended by successively applying the hash function to the last value and checking whether equal amount of computations results in the first value, i.e., the anchor.

Implementation

Our proof of concept is based on the HIP for Linux (HIPL) software bundle (version 1.0.4) [15], which we modified to implement our enhancements. For RADIUS experiments, we used FreeRADIUS 2.1.8. The implementation architecture is shown in Figure 3, where the red color depicts our modifications. In other words, two new logical entities were introduced, while the original HIP daemon was modified to include extra functionality without breaking the standard operation of it.

HIP daemon modifications mainly entailed introduction of new parameters and their processing within the corresponding messages. Some additional certificate processing

was also needed. The rest of the implementation architecture was based on the principle of making the existence of HIP transparent to RADIUS (depicted by the logical connection numbered 8 in the figure). Hence, it is easy to switch to use different RADIUS library, as the client part is contained within the RClient. Similarly, the client part can be easily made to directly access an external RADIUS server without HIP.

RClient and RDaemon can communicate with the HIP daemon using the internal communication mechanism available in HIPL. This way, HIP daemon is able to use RDaemon as a TTP selector. RDaemon is also responsible for encapsulation and decapsulation of the RADIUS messages, so that they can be conveyed using the HIP parameters. The various message sequences of the architecture are shown in Figure 4 (the numbers coincide with those in Figure 3).

Discussion

The important point in our work is how the different parts of the interaction are bound to the identities of the parties. First, the identity of a party is presented with a HIT, derived from a cryptographic public key. The negotiation offer and response are then signed with the corresponding identities, and as the signed data in the response also contain the anchor of the hash chain, the client binds itself to the evidence tokens. After all, with the assumption of irreversible hash function, no one is able to generate the values. One additional binding is provided for the traffic

HIP FORMS THE BACKBONE OF OUR IDENTITY-BASED APPROACH AND PROVIDES THE NEGOTIATION STEP, DURING WHICH THE KEYS FOR PROTECTING THE SUBSEQUENT SERVICE TRAFFIC AND BINDINGS TO THE NONREPUDIABLE EVIDENCE ARE ESTABLISHED.

when it is protected using the keying material derived during the handshake, which was authenticated with the used identities. Thus, every part of the system can be tracked to the identities of the participants, allowing us to ensure the accountability of the service provisioning. This way, the parties can be sure that they are communicating with the correct entity and that they have a suitable nonrepudiable evidence to resolve any disputes that might arise. They are also able to react during the service provisioning to any dishonest behavior. In other words, if the service does not keep receiving hash chain tokens, it can terminate the provisioning. Similarly, if the client does not receive the promised service, it can stop sending any more tokens.

As we used SPKI certificates to encode the authorizations and the commitments of the different parties, there is the question of revocation of certificates. For this, we adopted an approach where the authorizations are short lived, thus controlling the possible misuse window. This simplifies things, but one has to observe the tradeoff

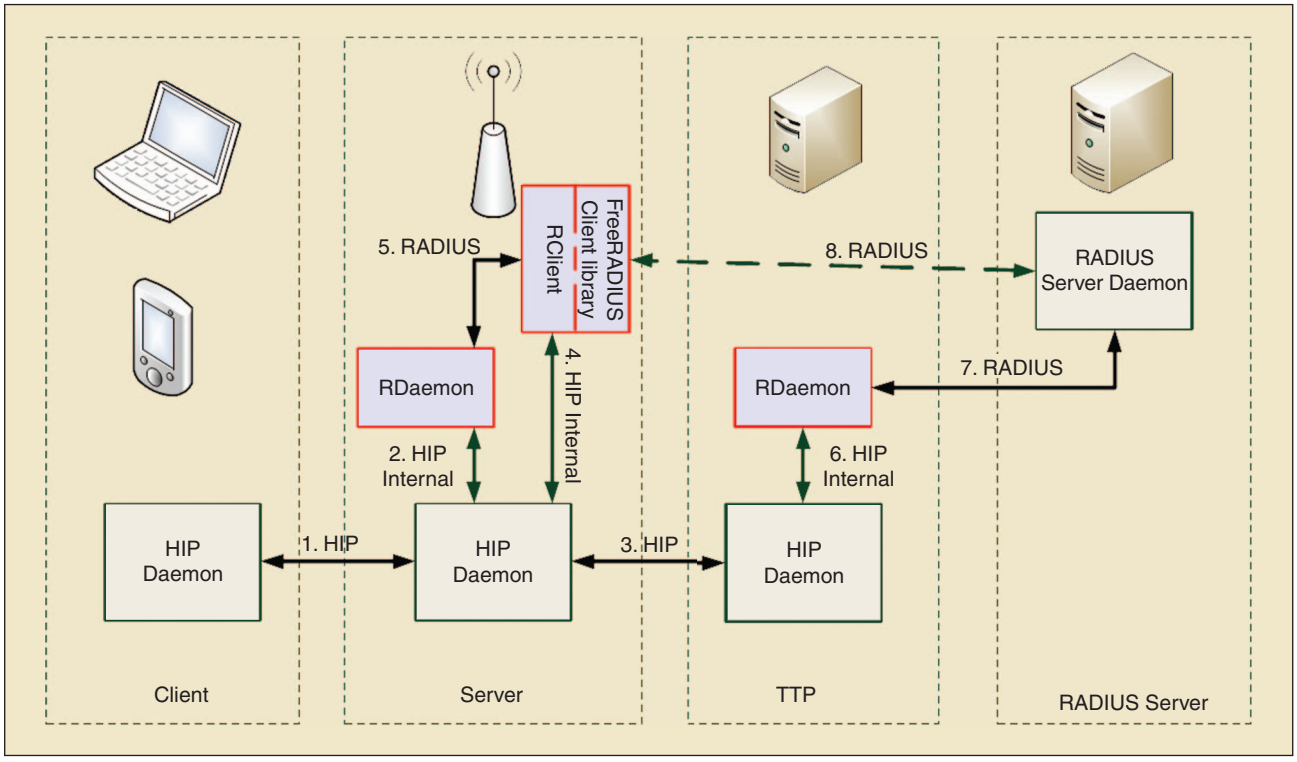


FIGURE 3 Implementation architecture.

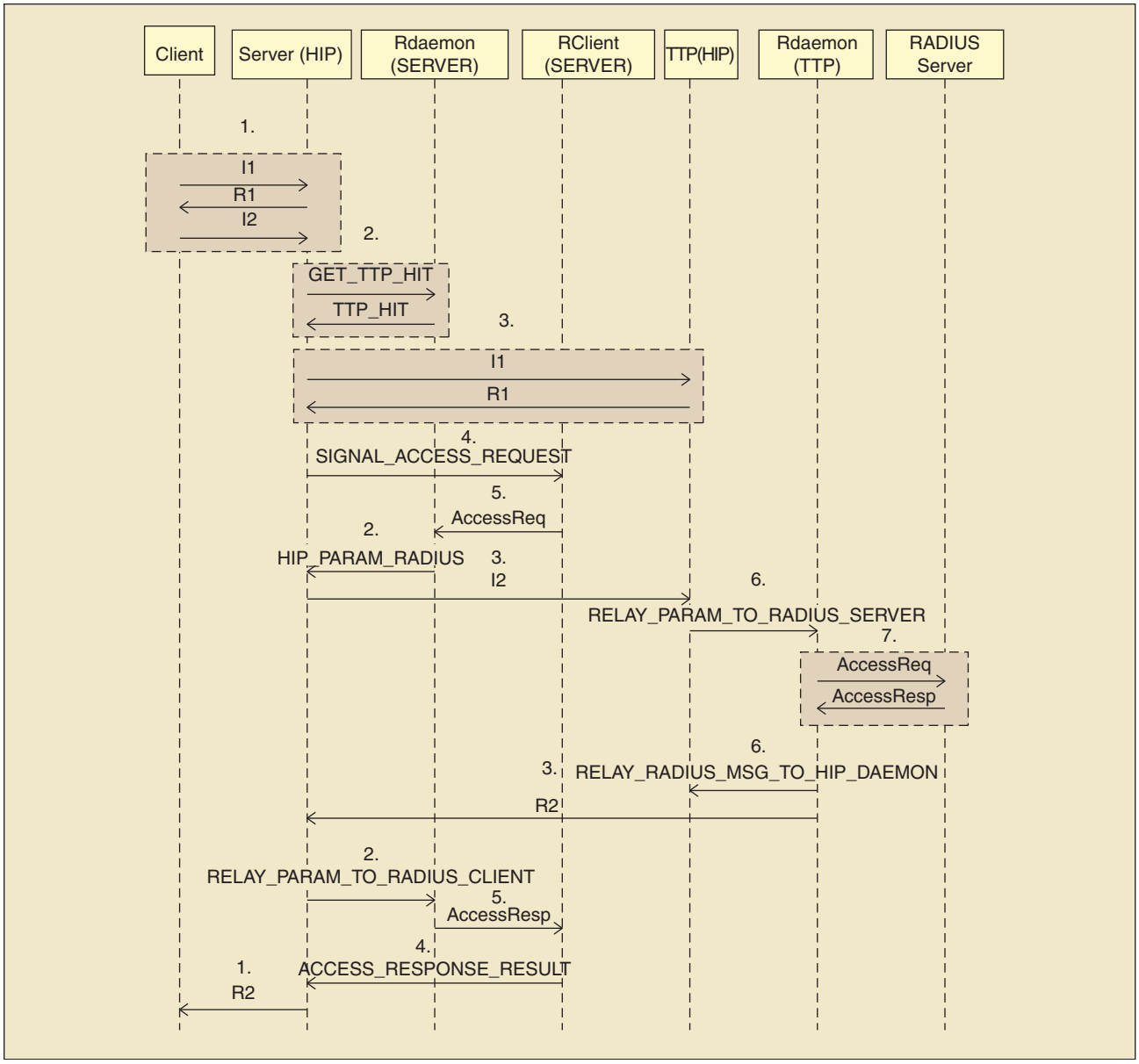


FIGURE 4 Message sequences for the implementation architecture.

between the amount of renewal signaling and certificate validity period.

The renewal issues are present with the hash chains as well, as they are of finite length, and the length is decided by the client at the time of negotiation. The client can use the offer validity time to estimate the needed length, but we also have the simple possibility of renewing the hash chain by binding the last and first values of the old and new chains to avoid running BEX again.

As HIP signaling is used to transfer the hash chain values acting as evidence tokens, our approach decouples the traffic from signaling. This way, it is possible to route the signaling via the home-operator network and give the home operator middleboxes the possibility to observe the consumption of service resources. Thus, one could optimize the data traffic and forego the step where it is routed through the home-operator network, as often seems to be the case with the cellular network.

While from the risk-management perspective the value of single token should be small, the granularity cannot be infinitely small. It is restricted by the creation and processing times of the update packets (and clock granularity). Thus, based on our experiments, one cannot expect the update granularity to be in the order of milliseconds, rather in the order of tens of milliseconds at most, although one needs to take into account the transmission delays as well. The server also has to consider how frequent updates it can handle, especially if it expects to serve several clients. However, one also faces the dilemma of lost values, i.e., how quickly one determines that the other party is

WE PRESENT AN AUTHORIZATION SOLUTION THAT TAKES INTO ACCOUNT THE IDENTITIES OF THE PARTIES INVOLVED IN THE SERVICE TRANSACTION.

dishonest. In terms of robustness, one ought to be willing to tolerate a couple of lost values, especially if environment is expected to be prone to packet loss. This is basically a policy decision, though.

From the performance perspective, our nonrepudiation enhancements add some tens of milliseconds of performance penalty. The typical HIP procedures still dominate the performance, though, as the base exchange runs in the order of several hundred milliseconds with our hardware. The inclusion of RADIUS authorization adds more penalty, but it is also affected by the networking delays. Naturally, for servers engaged in frequent handshakes and multiple connections, the performance can be severely impacted, but the same issues have to be tackled in the normal deployment of IPsec too [16]. Alternatively, one can also consider the benefits of employing elliptic curve cryptography in terms of better performance and smaller key sizes [17], even though it is not part of HIP standard yet. However, HIP is still years away from widespread deployment.

Conclusions

In this article, we have presented an implementation for enabling nonrepudiable network-level service usage by taking advantage of the characteristics of HIP. This approach can work as a building block for an accounting solution of future networks that takes into account the assurance needs of the user and service provider. Thus, the user can be certain that he/she only provides evidence for the resources he/she has used, and the service provider can make sure that it has legitimate claims to the compensation based on the nonrepudiable accounting records. This can be further coupled with the traditional RADIUS-based accounting systems.

The presented concept aims at providing service provision granularity that will make it possible to reduce the financial risk, i.e., it makes it easier for the partners to manage their risk in network-level scenarios. The implementation also demonstrates the benefits of HIP and an identity-based approach in binding different parts of the system in a secure way, i.e., partner identities, negotiation procedure, and the actual traffic.

Acknowledgments

This work was supported by Teknologian ja innovaatioiden kehittämiskeskus (TEKES) as part of the future Internet program of Tieto-ja viestintäteollisuuden tutkimus Oy (TIVIT); Finnish Strategic Centre for Science, Technology, and Innovation in the field of Information and Communications Technology (ICT).

Author Information

Seppo Heikkinen (seppo.heikkinen@tut.fi) received his M.Sc. degree in 1998 and licentiate of science degree in 2006 from Tampere University of Technology in Finland. He worked for the Finnish telecom operators during 1997–2005, and since 2006, he has been working in the Department of Communications Engineering for his Ph.D degree. His research interests include future network security architectures, identity-based security, and usable security.

Santeri Siltala (santeri.siltala@tut.fi) is an M.Sc. student and a research assistant. His research interests include nonrepudiable service usage architectures for future IP networks and identity-bound accounting solutions.

References

[1] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, Eds., (2008, Apr.). Host identity protocol, Internet Engineering Task Force Request RFC 5201. [Online]. Available: http://www.ietf.org

[2] R. Rivest and A. Shamir, "Payword and micromint: Two simple micropayment schemes," in *Proc. Int. Workshop Security Protocols*, Apr. 1996, pp. 69–87.

[3] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[4] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "SPKI certificate theory," Internet Engineering Task Force Request RFC 2693, Sept. 1999.

[5] C. Ellison, Ed., "Simple public key certificate," IETF Internet-Draft draft-ietf-spki-cert-structure-06 (expired), July 1999.

[6] J. Zhou and D. Gollmann, "A fair non-repudiation protocol," in *Proc. 1996 IEEE Symp. Security Privacy*, May 1996, pp. 55–61.

[7] N. Asokan, "Fairness in electronic commerce," Ph.D. thesis Univ. of Waterloo, Waterloo, ON, Canada, 1998.

[8] S. Herda, "Non-repudiation: constituting evidence and proof in digital cooperation," *Comput. Standards Interfaces*, vol. 1, no. 1, pp. 69–79, Jan. 1995.

[9] D. Mitton, M. St. Johns, S. Barkley, D. Nelson, B. Patil, M. Stevens, and M. Stevens, "Authentication, authorization accounting: Protocol evaluation," Internet Engineering Task Force Request RFC 3127, June 2001.

[10] P. Louridas, "Some guidelines for non-repudiation protocols," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 5, pp. 29–38, Oct. 2000.

[11] J. Zhou and K. Lam, "Undeniable billing in mobile communication," in *Proc. 4th ACM/IEEE Int. Conf. Mobile Computing and Networking*, Oct. 1998, pp. 284–290.

[12] H. Tewari and D. O' Mahon, "Multiparty micropayments for ad hoc networks," in *Proc. IEEE Wireless Communications and Networking Conf*, Mar. 2003, pp. 2033–2040.

[13] M. Blaze, J. Ioannidis, S. Ioannidis, A. D. Keromytis, P. Nikander, and V. Prevelakis, "TAPI: Transactions for access public infrastructure," in *Proc. Personal Wireless Communications (PWC2003)*, Sept. 2003, pp. 90–100.

[14] S. Li, et al., "Fair and secure mobile billing systems," *Wireless Personal Commun.*, vol. 51, no. 1, pp. 81–93, Oct. 2009.

[15] Helsinki Institute of Information Technology. (2010, Oct.). HIP for Linux (HIPL). http://infrahip.hiit.fi/

[16] C. A. Shue, M. Gupta, and S. A. Myers, "IPSec: performance analysis and enhancements," in *Proc. IEEE Int. Conf. Communications*, June 2007, pp. 1527–1532.

[17] O. Ponomarev, A. Khurri, and A. Gurtov, "Elliptic curve cryptography (ECC) for host identity protocol (HIP)," in *Proc. 9th Int. Conf. Networks*, Apr. 2010, pp. 215–219.

[18] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. Internet Engineering Task Force Request For Comments 5280, May 2008.

[19] International Telecommunication Union, "X.680—OSI networking and system aspects—Abstract Notation One (ASN.1): Specification of basic notation," *ITU-T Recommendation X.680*, July 2002. ***VT***

Publication P5

Heikkinen S., “Security and Accounting Enhancements for Roaming in IMS”, in *Proceedings of The 6th International Conference on Wired / Wireless Internet Communications (WWIC08), Lecture Notes in Computer Science 5031*, Tampere, Finland, May 2008.

Publication P6

Heikkinen S., "Establishing a Secure Peer Identity Association Using IMS Architecture", in *Proceedings of The Third International Conference on Internet Monitoring and Protection (ICIMP08)*, Bucharest, Romania, Jul 2008.

Establishing a Secure Peer Identity Association Using IMS Architecture

Seppo Heikkinen
Tampere University of Technology
firstname.lastname@tut.fi

Abstract

The advent of ubiquitous computing and the convergence of the heterogeneous networks provide new opportunities for the new players to enter the operator market. While network access will be available everywhere, the multitude and diversity of the access operators makes it hard to rely on the old paradigms of static operator relationships guaranteeing the identity of the users end-to-end. Therefore, there is need for mechanisms that allow the endpoints get assurance about the identity of their counterparts. This paper investigates the possibility of taking advantage of IP Multimedia Subsystem (IMS) in a roaming scenario to signal the needed identity parameters between two communication endpoints, which are basing their trust evaluation on their own home operators. This allows establishing a Host Identity Protocol (HIP) style identity association, which can be used to protect any subsequent communication between the same entities.

1. Introduction

While the future networks develop and become more dynamic in nature, the security requirements are even more critical in this new ubiquitous environment. As the envisaged ambient visions become reality and the technological development allows practically anybody to provide access services of their own, the amount of mini operators increases. Thus, the operator relationships may no longer be based on pre-established roaming agreement, but they are created on the fly. Such diverse environments require more assurance about the identities of the users and statements, which ensure the liability of the actions taken by them.

The users may have trust to their own operators, with whom they have billing relationship and who provide the user with authentication infrastructures, such as those based on the existence of Subscriber Identity Module (SIM) cards. In essence, these operators assume the role of identity providers. The large operators also have incentive to tie their customers more tightly to their own service provisioning architectures in order to be able to provide more profitable multimedia services. For instance, the operators are currently busy investigating the possibilities of IP Multimedia Subsystem (IMS) deployment.

In this paper we investigate the possibility of taking advantage of IMS architecture to establish a trust between two end points in a roaming scenario, where the visited access network may not be entirely trustworthy. In essence, this means establishing an identity association so that the parties can have operator provided assurance regarding the used identities. The home operators of the users are assumed to have established a typical roaming agreement, which dictates the liabilities of the parties and the necessary security association. The approach adopted here leans on the ideas developed for Host Identity Protocol (HIP). Thus, it is assumed that every entity is in possession of a secure cryptographic identifier, which provides secure naming through a proof of possession. So, in essence, we suggest using IMS infrastructure to carry the necessary initial signalling needed to enhance the trust established in a typical HIP exchange. Therefore, this allows local trust decisions and does not rely on the existence of global Public Key Infrastructure (PKI). Naturally, this is logically very close to a typical PKI setting, where we have operators acting as Certificate Authorities (CA), i.e. trust roots, which are willing to cross-certify each others.

The difference to the typical IMS setup is that the envisaged usage environment is not static in terms of relationships to the access operators, i.e. visited network. Rather it is expected that the relationships are established in a dynamic fashion and there is more uncertainty regarding the security of the en route network elements. While it is true that this at present does not seem so realistic scenario, but it is expected to be more likely in the future ubiquitous environment.

This paper is organised as follows. In the next section we briefly go through the basics of HIP. The third section provides an overview for IMS

978-0-7695-3189-2/08 $25.00 © 2008 IEEE
DOI 10.1109/ICIMP.2008.8

architecture. In the fourth section we sketch our high level architecture and the section after that gives more details to the employed messages. The sixth section provides analysis for the employed security measures. The seventh section gives some direction for future work and the next section concludes the paper.

2. HIP basics

HIP proposes new identity layer architecture for the Internet [1]. It positions itself between the network and transport layers and aims at solving the dual nature problem of the IP address. That is, currently it functions as an end point identifier and a locator. HIP provides means to identify the end points irrespective of its topological location, thus providing better solutions for mobility and multihoming issues. HIP also provides key negotiation capabilities with the help of Diffie-Hellman key exchange and simple denial of service (DoS) protection through a puzzle mechanism.

HIP hosts are identified by their cryptographic identifiers, which basically rely on the existence of a public key pair. By applying hashing to the public key one is able to form a concise representation of this identity, Host Identity Tag (HIT), which is more suited for protocols. Currently, as it is 128 bits long, it also has IPv6 interpretation [2]. During the protocol run proof of possession of HIT is provided.

HIP association is created through a four way handshake mechanisms, so called base exchange (see Fig. 1). The first message, called I1, is basically just a trigger for the responder to start the handshake. The responder replies with R1 message, which is precalculated in order to mitigate DoS concerns and contains identity information, i.e. the public key, and the parameters it wishes to use for this association. The information is signed with the private key, so it proves that it is in the possession of the correct key. This does not yet store any state and the responder expects to receive in the next message a solution to the puzzle it provided. The initiator provides the solution in I2 message, which also includes its own session parameters and identity information along with the corresponding signature. As the key material exchange is complete at this point, it is possible to also encrypt some information, like the identity, with the created keys to provide privacy protection. The last message, R2, which concludes the exchange, assures to the initiator that it is indeed talking to a live responder and it has knowledge about the session key. After this HIP association has been established. Further communication can be protected, for instance, with IPsec using the created keying material [3].

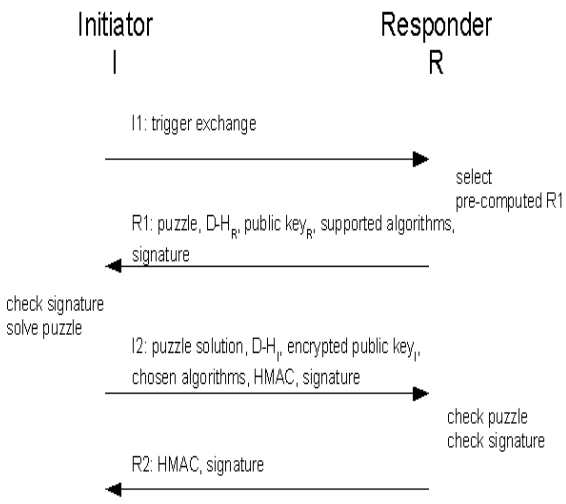


Fig. 1. HIP base exchange.

3. IMS overview

IP Multimedia Subsystem (IMS) is architecture for delivering multimedia services on IP based networks [4]. This service architecture was developed by 3GPP and while it is envisaged to be access technology agnostic, it is so far deployed on 3G cellular networks.

From the trust perspective the security of IMS is based on the fact that the communication is expected to be taking place between trusted entities [5]. In other words, the operators that exchange signalling have pre-established agreements, which define the limits of their interaction. This also includes the security measures, such as integrity and confidentiality, used to protect the connection between them. It is also expected that the operators keep their internal network secure, so that all the network elements are trusted entities. Hence, it is not mandatory to secure the connections between one's own network elements. Due these assumptions the home operator, for instance, relies on the statements made by the access operator regarding the user actions and its identity, even though these statements do not enjoy any real protection.

Functionalities of IMS are actuated with the help of Session Initiation Protocol (SIP), which is perceived as a general signalling protocol for establishing and controlling the sessions between the communicating entities [6]. Public identities are expressed with email-like Address of Record (AoR), which gives indication of the home domain of the user, although the real contact identity might be more specific depending, for

instance, on the currently used device. SIP is transaction oriented and works in a hop-by-hop fashion, so each network element, or proxy, on the path can make its own changes to the control information in the messages in order to provide either additional services or ensure correct routing. This way the network can also react to any quality of service requirements the session might require, hence complete end-to-end confidentiality is not desired. SIP is basically a simple text based protocol and the individual messages contain a header and a body section, much in the way as in HTTP. Headers include most of the control information and the body section usually contains information regarding the information content the parties are negotiating about. It could be, for example, used to describe the media session to be negotiated and use a different protocol, such as Session Description Protocol (SDP). In theory, however, SIP message body could contain any other type of content as well. SIP itself enjoys very little protection in IMS architecture due the previously mentioned fact that the communication is expected to take place between trusted entities. In basic SIP architecture, though, one could use, for instance, S/MIME, although it does not define how to convey trust between the entities.

The simplified architecture is depicted in Fig. 2 in terms of session establishment between two users. The more detailed architecture can be found in [4]. The first SIP contact point for the user is Proxy Call Session Control Function (P-CSCF). It is responsible for finding the next contact point, which in the case of home network might be Serving CSCF (S-CSCF) or in case P-CSCF is in the visited network, then Interrogating CSCF (I-CSCF) of the home network that the home network uses as a published entrance point to its network. P-CSCF could also use other border elements, such as Interconnection Border Control Function (IBCF), to take care of the communication with the other networks. P-CSCF can also interact with transport level entities, so it can set policies for the handling of the data traffic of the user. I-CSCF is also responsible for finding an appropriate S-CSCF in the home domain to serve the roaming user.

S-CSCF is the "work horse" of IMS system as it is responsible for authenticating the user with the help a challenge-response procedure called Authentication and Key Agreement (AKA), which also registers the SIP identity of the user. It does this in cooperation with the Home Subscriber Server (HSS), which contains all the subscriber information. S-CSCF is also in the path of every SIP message the user sends or receives, so it can redirect the messages to the other networks or the appropriate application servers (AS) as dictated by the profile of the user. This allows it to provide accounting information, as well. In subsequent communication the intermediary proxies do not necessarily need to take part in the signalling, so, for instance, P-CSCF and S-CSCF could be the "neighbouring hops". There are also other network elements to take care of the media processing and interaction with other networks, such as the legacy telephone systems, but they have been left out of the scope of this paper.

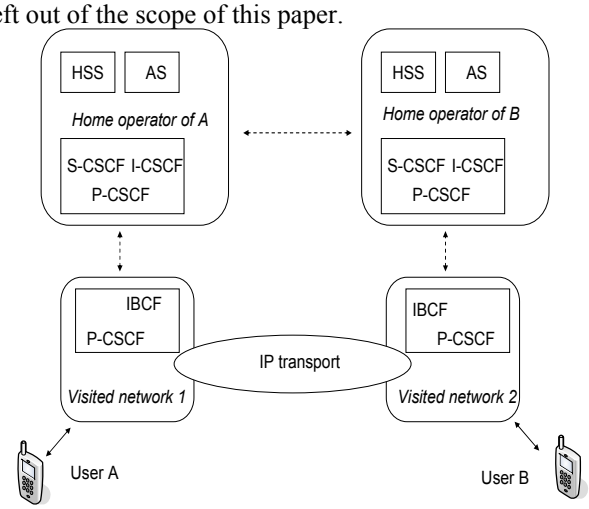


Fig. 2. Simplified IMS architecture.

4. High level overview

The flow of events is depicted in Fig. 3 and goes as follows. In the first SIP INVITE message the UserA sends its identity information along with the association specific configuration information, such as keying information. In practise, in the envisaged architecture this is a HIP packet. The identity information includes HIT and the relevant public key information. The keying information uses Diffie-Hellman exchange as per typical HIP handshake. The difference to the typical base exchange is that there is no triggering, so basically the roles are reversed and the R1 message is seen as the one starting the handshake instead of I1. This does not need to contain any puzzle scheme, though, because if the sender is not known by the signalling network, then the message will not be forwarded to the final recipient. After all, it is expected that the host has already registered with the home operator using AKA procedure. The message is targeted to a certain SIP identity, for which HIT may not be known. In fact, if HIT is known "reliably" then the negotiation does not need to go through the operator infrastructures as there is no need to get guarantees for the used identities. The contents and the relevant SIP headers are protected with a signature.

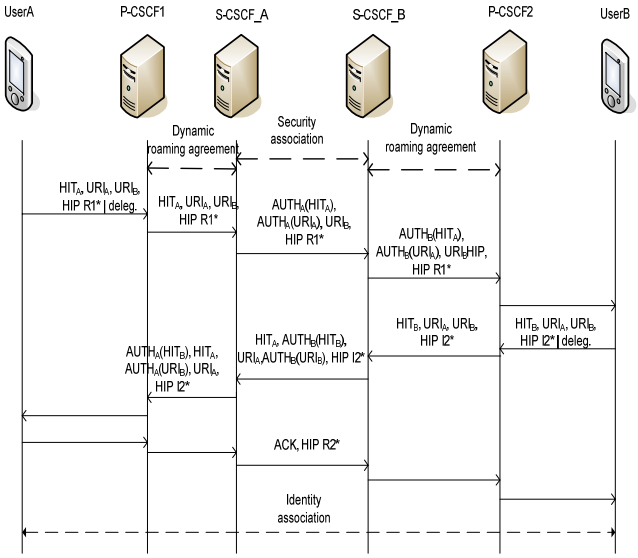


Fig. 3. Message flows between the network elements.

In the next phase P-CSCF1, which might be under different administration than the home network, does not do much else than to forward the message, although it can include the P-Asserted-Identity to indicate its own notion of the identity of the user [7]. After all, P-CSCF1 should be aware of the user identity based on the in the initial registration done originally between the two entities. Some charging specific information can also be added as per typical IMS procedure [8]. However, the reliability of this information depends on the existence of security association between the access and home operators as it is generally expected that the IMS charging information is exchanged only between trusted networks. This might not always be the case in the scenario we are envisaging. One could, though, employ solutions that provide non-repudiation for the service usage [9].

When the S-CSCF_A, i.e. the one in the home network of UserA, receives the message, it can check the correspondence of the registered user identities. In other words, it is assumed that the user has previously registered the identities it is using on this connection and that the operator has authority over the used identities. After this and the verification of the signature, S-CSCF_A can sign the user's HIT and the expressed application level identity as an indication of trust to those identities along with all the other immutable headers (see later section). Based on the receiver's AoR, the message is forwarded to the home operator of that entity.

When S-CSCF_B, i.e. the home network of the UserB, receives the message, it can check that the identities are assured by the other operator, with whom it has a roaming agreement and has exchanged identities. Thus, it is willing to forward traffic from this operator to its own subscribers. S-CSCF_B proceeds in including its own assurance to the identities of UserA as a token of the trust it has on the established agreement. This takes in the form of signature as in the previous step.

In the last phase UserB receives the message and can check that there is an assertion made by its own home operator. It is assumed that the user is in the possession of the identity of its home operator. Thus, it can note the correspondence between the two identities of UserA and send the response with its own configuration information and the relevant identities. Within the HIP packets is also the authentication code calculated from the keying material, so that the UserB can prove to be in the possession of the session key.

The procedure then works to the opposite direction in similar fashion. In other words, the home operator of UserB assures the identities and based on the transitivity, the home operator of UserA asserts the received identities.

After this UserA can acknowledge the transaction with ACK message as in typical SIP transaction and includes also to the HIP packet an authentication code calculated with the exchanged keying information. After this the parties can switch to communicating directly on IP layer with the negotiated session parameters as they are in the possession of each others assured identities. Further traffic could be protected, for instance, with IPsec [3].

5. Messages

The first message sent by UserA takes advantage of the principles of the SIP Identity scheme [10], although with the modification that the signature is done at this point by the end entity and not the authentication server. This is indicated with P-End-Identity-Info, which identifies the identity, and P-End-Identity, which contains the corresponding signature. There is also need for additional field for indicating the identity, which the UserA wants to communicate to UserB to be used in conjunction with the identity association, i.e. P-End-Pub-Identity in the Fig. 4. While this could be derived from the From header, similar identity information cannot be included in the same field in messages coming from the opposite direction, because those fields are not expected to be altered during the transaction, if normal SIP INVITE semantics are to be honoured [6]. After all, the final peering partner might be some other identity due call

forwarding or some other similar functionality. The secure identity, i.e. HIT in this case, is expressed in P-End-Identity-Info header.

The actual keying information and other configuration specific information are in the body section using typical HIP packet syntax. However, one additional requirement is set for the signed nonce (echo) parameter used within the packet: in order to bind it to the current transaction, it uses the same call id as used in the corresponding SIP header. The encoding of the packet could be binary or base64, although some implementations might have difficulties with the binary content. However, it has the benefit of making the message compact, thus making more likely to fit the message inside a UDP datagram instead of having to switch using TCP as required by SIP RFC in the case of too large datagrams [6]. With all the options, this might be hard to avoid, though.

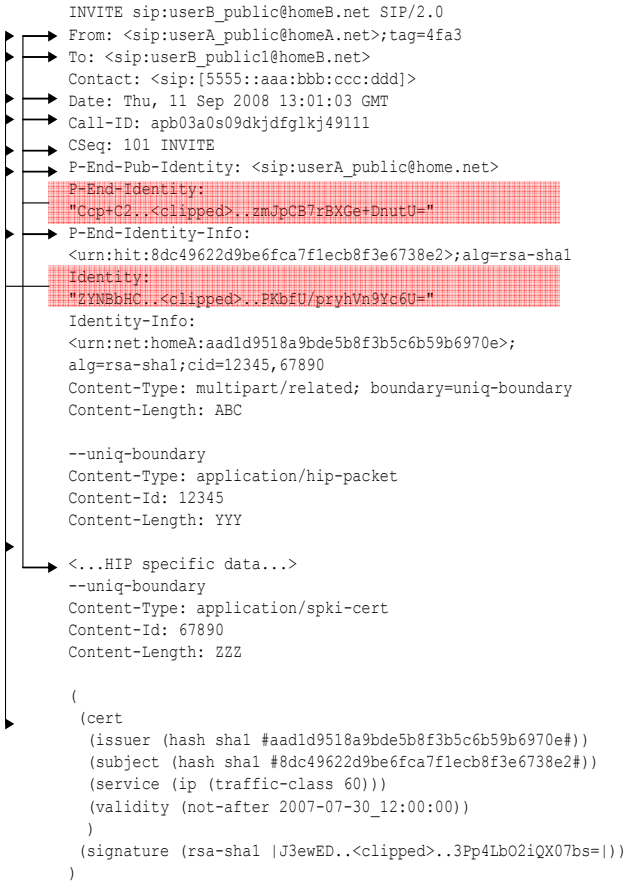


Fig. 4. Example of SIP message travelling from operator A to operator B and the coverage of the used signatures (some header values shortened).

Instead of having the HIP parameters in the body, the user has the option of including a delegation certificate, which authorises another network element to take care of the HIP negotiation, much like depicted in [11]. While this option does not provide end-to-end security, it could be required in certain cases. One such case is requirements set by the regulatory bodies that state that there has to be possibility of lawful interception for the authorities [12]. Thus, this kind of arrangement could be then handled, for instance, by Gateway GPRS Support Node (GGSN), which has interface with the first proxy element and contact point of the user, P-CSCF, and which also acts as the IP traffic gateway of the user.

The message, which is forwarded by the home operator of UserA, is attached with the signature of the operator stating that the identities belong to a trusted user. In addition, it could contain other information, which authorises only certain actions. That is, it might be that the operator wants to restrict actions of the user it wants to be liable for. This can be done with the help of a suitable attribute certificate or Simple PKI (SPKI) certificate as done in the Fig. 4. It is also possible just to provide a URL and hash of the certificate in order to save space in the message itself. The signature of the operator is extended over the identities of the user, i.e. both the HIP and SIP levels. This should also be extended to the certificate section, so that it is not possible to snip away the certificate in order to create confusion regarding the granted user rights. Some additional headers are also signed in order to guarantee the authenticity of them. An example of SIP message is given in Fig. 4, although some of the unrelated headers are left out in order to make the picture more compact. Arrows in the figure indicate the parts, which are taken into the corresponding signature calculation. An example of what the delegation certificate issued by UserA might look like is shown in Fig. 5.

```
(
 (cert
  (issuer (hash sha1 #8dc49622d9be6fca7f1ecb8f3e6738e2#))
  (subject (hash sha1 #<..middlebox hit..>#))
  (session (call-id apb03a0s09dkjdfglkj49111))
  (privileges (negotiate-hip))
  (validity (not-after 2008-08-11_14:00:00))
  )
 (signature (rsa-sha1 |<..signature here..>|))
)
```

Fig. 5. Example of an SPKI certificate for delegating HIP negotiation to a middlebox (such as P-CSCF).

When the other operator receives the message, it can replace the operator level signatures with its own and also provide a certificate. While everything could

be kept in the same message, in order to keep the size constrained one should take away the "unneeded" parts. After all, the user has his trust on his own operator, not to some other, potentially unknown operator. Of course, this leads to a slight misconception that the operator trusts the subscriber of another operator, when in truth the trust is only transitive. However, the other operator has accepted the liability of the user actions, thus, in essence, the trust can be placed on the subscriber of the other operator, naturally within the limits of the established roaming agreement. Dishonestly acting home operators are another issue, but in such case the subscriber is in trouble anyway.

HIP specific parts can be included in the messages either using binary or base64 encoding. This is similar to the way Multimedia Internet KEYing (MIKEY) exchange has been envisaged to be used with SIP [13]. One might ask whether it is reasonable to duplicate some of the information that is available in the other parts of the SIP message, but this way the whole HIP packet can be fed to the HIP processing without major changes, providing better compatibility with the "pure" HIP implementations. There might also be a question of whether it would not be better just to use MIKEY for key exchange, but the idea here is to specifically create a HIP compliant identity association between the parties that can be later used in their mutual communication to establish a certain level of assurance.

6. Analysis

The scheme is based on the idea of using the existing trust relationship between the operators to establish trust between two end entities. In essence, this means corroborating the binding between two different level identities, i.e. HIP and application level. When the original message is signed by the end entity, it assures that it is the sender of the message and it is the originator of the content and certain header parts, so that the access operator, for instance, has not been able to tamper with them. As the user has previously registered with the home operator, the operator can check the correspondence of the identities and augment the message with its own assertion. Because there is a pre-established agreement between the home operators, the other operator trusts the assertion made by the first operator. Hence, it can make a similar assertion to its own customers. The users are in the possession of the identities of their operators, so they are able to make the necessary checks to ensure that the assurance is provided by the correct entity. Additionally, they have a shared secret with their home operator that allows them to run AKA procedure during the registration phase. So the operator assured liability is an important distinction to completely decentralised web of trust kind of approaches.

The receiving end entity can be certain that the received parameters are related to the sending entity due to the existing signature. Trustworthiness comes from the fact that the operators on the path have asserted the correspondence of the identities, even though the receiving entity may not see any signature made by the first operator.

The signature of the user protects the From, To, Date, Call-Id, Cseq, P-End-Pub-Identity, P-End-Identity-Info, and the primary body section, which might be delegation or HIP parameters. While P-End-Pub-Identity is also signed by the operator, it is worthwhile to sign it by the user as well, so that the user has explicitly stated, which identity it wishes to use with later communication with the other peer. Signing Date makes it harder to try replay attacks and Call-Id identifies the current transaction. When Cseq is included in the signature, it is not possible to try to forge the used SIP method, i.e. like trying to change INVITE to BYE.

In addition to the above, the signature of the operator protects the possibly attached operator certificate regarding the user privileges. The main idea of this signature is to bind the used identities so that the operator gives assurance that there is a legitimate binding between those two level identities. The reason for not including the user signature to the calculation is partly performance related, i.e. one does not need to verify two signatures in order to make sure that the operator signature is valid. However, as the operator also takes into its own calculation the critical headers, there is no fear of compromised headers.

The reason for including the operator certificate to the header level signature is to make sure that it cannot be removed from the message. This might give unnecessary broad privileges to the user, when the intention of the operator was to limit them to a certain subset. The signature in the certificate itself might be considered to be duplication of information, but this way the certificate can be used on its own on other protocol levels as well.

One might ask why not just use operator issued certificates and cross-certification across operators to establish the association between the end entities. While it might be an option as well, this way one does not need to provide complex certificate management procedures nor transmit certificate chains. The requirement of having unfragmented HIP packets sets limits to the amount of information that can be put

there (see [9]), so having a pure HIP solution with certificates is size constrained. However, employing a HIP compatible solution enables one to take advantage of the suggested locator/identity split.

7. Future work

The architecture outlined here is so far only a sketch of the possible solution and further work is needed to investigate the implementation and scalability aspects. For instance, it is not likely that this approach would be feasible for every possible connection setup due the inherent SIP efficiency issues in IMS environment. However, this can be used for select scenarios to enhance the trust to the employed identities. In a typical case of multiple new connections in a less critical environment it is possible to just use HIP for association establishment and it can take an opportunistic approach for the trust establishment, if no other option is available.

More work can also be done for providing additional mobility support from the infrastructure. In other words, it might be possible for the end entities to provide the contact information in form of HITs instead of IP addresses and then expect the core network elements to provide rendezvous kind of service, as depicted in [14]. Using HITs with SIP is actually already investigated in [15], but it is used purely as a mobility solution.

8. Conclusion

In this paper we have presented an architecture, which aims at enhancing a typical HIP exchange by leveraging the trust relationships and the signalling mechanisms of IMS in a roaming environment, where the conduct of the access networks can be dubious. By taking advantage of the trust assessment made by the operators, the end entities are able to get a notion about the trustworthiness of the identities of the peer party. This notion can be used to secure the subsequent direct communication between these parties. The presented scheme also makes it possible to include HIP parameters directly into the SIP messages, so the HIP handshake is taking place along with the SIP INVITE transaction, which provides the identity assurance.

The presented work is still a high level sketch and further work is needed to experiment with the scalability issues. However, it gives an interesting starting point for considering the future network architectures and the possibilities to take advantage of the envisaged identity schemes and existing trust relationships without having to deploy global PKI.

References

[1] Jokela P. (Ed.). Host Identity Protocol. Internet Draft draft-ietf-hip-base-10, work in progress. Oct 2007.

[2] Nikander P., Laganier J., Dupont F. An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID). IETF RFC 4843. Apr 2007.

[3] Jokela P., Moskowitz R., Nikander P. Using ESP transport format with HIP. IETF Internet-Draft draft-ietf-hip-esp-06, work in progress. Jun 2007.

[4] 3GPP. IP Multimedia Subsystem (IMS). 3rd Generation Partnership Project Technical Specification. TS23.228 V8.1.0. June 2007.

[5] 3GPP. Security architecture. 3rd Generation Partnership Project Technical Specification, TS 33.102 V7.1.0. Dec 2006.

[6] Rosenberg, J. et al. SIP: Session Initiation Protocol. IETF RFC 3261. June 2002.

[7] Jennings C., Peterson J., Watson M. Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks. IETF RFC 3325. Nov 2002.

[8] Garcia-Martin M., Henrikson E., Mills D. Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP). IETF RFC 3455. Jan 2003.

[9] Heikkinen S. Non-repudiable service usage with host identities. Proceedings of Second International Conference on Internet Monitoring and Protection. Jul 2007.

[10] Peterson, J., Jennigs, C.: Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP). IETF RFC 4474. Aug 2006.

[11] Koponen T., Gurtov A., Nikander P. Application mobility with Host Identity Protocol. Proceedings of Network and Distributed System Security Workshop '05. Feb 2005.

[12] The Council of the European Union. Council resolution of 17 January 1995 on the lawful interception of telecommunications (96/C 329/01). Official Journal of the European Communities. Nov 1996.

[13] Arkko J., Carrara E., Lindholm F., Naslund M., Norrman K. MIKEY: Multimedia Internet KEYing. IETF RFC 3830. Aug 2004.

[14] Laganier J., Eggert L. Host Identity Protocol (HIP) Rendezvous Extension. IETF Internet-Draft draft-ietf-hip-rvs-05, work in progress. Jun 2006.

[15] So J.Y.H., Wang J., Jones D. SHIP Mobility Management Hybrid SIP-HIP Scheme. Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. May 2005.

Publication P7

Heikkinen S., Silverajan B., ”An Architecture to Facilitate Membership and Service Management in Trusted Communities”, in *Proceedings of The International Conference on Computational Aspects of Social Networks (CASoN 2009)*, Fontainebleau, France, Jun 2009.

An Architecture to Facilitate Membership and Service Management in Trusted Communities

Seppo Heikkinen and Bilhanan Silverajan
Department of Communications Engineering
Tampere University of Technology
Tampere, Finland
Firstname.Lastname@tut.fi

***Abstract*—Ubiquitous connectivity today allows many users to remain connected regardless of location with various kinds of communities. This paper studies challenges in building trusted communities that encompass both new users as well as users already possessing credentials from other well known connectivity providers, federations, content providers and social networks. We postulate that trusted communities are initially created as a means to access some services, but become enriched with user created services. We present an architecture aimed at managing the complexity of service composition, access as well as guarantees of authenticity. Since users possess multiple credentials from various identity providers, we address this in our architecture from the service access perspective. In addition, our model explicitly takes into account cases where users may temporarily be granted access to a community's services based on recommendations from existing members.**

Keywords-online communities; identities; trust and service management

I. INTRODUCTION

A Trusted Community is a community whose infrastructure is capable, if necessary, of providing a high level of confidence to its participants with regards not only to the identity of its members, but also the authenticity of all the services and content that may exist within that community. Users possess a myriad of different devices that enables them to remain connected in the face of today's pervasive networking culture. Indeed, the ready availability of high speed wireless broadband-like connections today have spurred levels of participation in the Internet where users would demand different levels of trust in their interactions with other users and services. Community and public networks created from surplus bandwidth available via excess capacity in local loops (homes, campuses, hotspots) are one example of Trusted Communities, which require a stronger assurance that the service offered is genuine. User identification may also be enforced depending on whether the user is participating in offering surplus bandwidth. This provides a good guarantee against malicious behaviour for all the stakeholders involved, including the operators running the networks.

Today, social networks provide a quick means for groups of friends to stay in touch with one another, going beyond simple e-mail and having web pages towards instant messaging and blogs. As usable social networking platforms began arriving, the technologies collectively termed as Web 2.0 began transforming the Web into a collaborative space for sharing content and enabling multimedia communication. At the same time, they began to provide lightweight programming platforms to enable rapid development of web-based widgets and applications. Rapid adoption of social networking and Web 2.0 platforms worldwide have led to the creation of large online communities as well as service providers who are beginning to open up aspects of their platforms to allow for interoperability and service mash-ups.

Although social networks provide good platforms for interactions and exchanging applications and related content for their respective users, various factors often conspire to pose a hindrance towards a true exchange services by users of social communities. Such factors may include a need imposed upon all users to belong to the same network, invasive techniques for extracting existing user and address book information, ambiguous privacy agreements and data mining, coarse grained access control to media, the inability to form smaller private groups to share content or verify how safe any executable applications being shared are. This consequently again creates a need for either a closed group to interact with all users known to the others, or a wider social community in which incoming users can be vouched for by existing community members.

This paper presents this idea of a trusted community, in which each user or service is well-known and is authenticated into the community via a variety of predefined methods. Services within such a community can be composed into larger offerings without sacrificing the authenticity of individual components and can be offered by both a predefined community platform provider or by users themselves. We also look at how the creation of such service-based communities can be facilitated by middleware.

We undertake to give a broad view of services, classify the different kinds of services that may exist in a community, ranging from network-level services to high-level, web-based widgets and applications. At the same time, we take the growing popularity of social networks and platforms into account and attempt to leverage their APIs for managing authenticity, trust and federated identities of a Trusted Community's membership.

Section II presents our conceptual perception of communities, services and users in a mathematical model. Section III provides a reference architecture while Section IV

978-0-7695-3740-5/09 $25.00 © 2009 IEEE
DOI 10.1109/CASoN.2009.16

outlines implementation and interaction scenarios. Future research challenges that we anticipate are finally presented.

II. Conceptual Model & Terminologies

We consider a community to be initially service-based rather than user-based. That is, a community is not initially defined only by the group of users in it nor by their social or professional interests. Instead a community is initially a static abstract space made possible by the existence of one or more services without any users. Given a possibly infinite set of existing services ES, we then define the set of abstract communities *AC* to be a subset of *ES*. Each abstract community exists solely by virtue of an initial subset of services *IS*. We represent these relationships as:

$$AC \subseteq ES, \text{ where } ES = \{es_1, es_2, \ldots, es_i, ..\}, \quad AC = \{ac_1, ac_2, \ldots, ac_i, ..\} \tag{1}$$

$$IS = \{is_1, is_2, \ldots, is_i, ..\}, \text{ where } \forall is_i \in ES \tag{2}$$

We also define a possibly infinite set of users, *U*. When an initial user indicates an interest in using any service that has been tagged as falling within a certain community, the community then becomes a concrete instantiation with a well-known identifier, with the user then joining the named community as a member in order to use its service. Alternatively, the member can choose to enrich the community's pool of services with more service offerings. The set of community services, *CS* thus includes both Initial Services *IS* (services that initially defined the abstract community space) together with User Services *US*, a dynamically changing set of services that users contribute to the named community. Therefore, our conceptual model defines named communities in the set *NC* as a function of a 2-tuple, represented with the following rule:

$$NC = F(U, CS), \text{ where } U = \{u_1, u_2, \ldots, u_i, ..\}, \quad CS = IS \cup US \tag{3}$$

The named community consequently differs from its initial conception of an abstract community, in that the static abstract space comprised of a fixed set of services is transformed into a virtual space that dynamically grows or contracts based on the number of users joining and leaving and subsequent service offerings in the pool. This is represented in Figure 1.

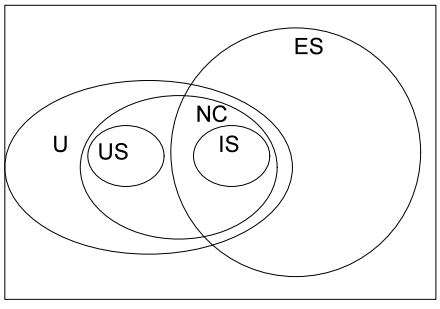


Figure 1. Conceptual Relationship Model

Each community can choose to have an arbitrary number of members. Additionally, users can belong to one or more communities, with various levels of overlapping or nesting relationships among these communities.

We assume that users will carry multiple credentials that would authenticate them to different kinds of social communities. We can also assume at least some of these communities would be strict subsets of other communities. Consequently, if a user becomes a member of a nested community, we also assume that he simultaneously becomes a member of the nesting community. If we define a *join* event,

$$join\ (u_q, nc_{ij}),\ q \in 1 \ldots n,\ \forall nc_{ij} \in NC$$

at any discrete point in time *t*,

$$t\ (join\ (u_q, nc_i)) \Rightarrow t\ (join\ (u_q, nc_j)),\ iff\ (nc_i \subseteq nc_j) \tag{4}$$

However, the converse does not hold true. In other words,

$$t\ (join\ (u_q, nc_i)) < t\ (join\ (u_q, nc_j)),\ iff\ (nc_i \supset nc_j) \tag{5}$$

This will therefore give us an ordered sequence of events where a user initially joining a named community nc_i would not automatically be a member of the nested community nc_j, needing to subsequently make an explicit join at a later discrete point in time. On the other hand, no such restriction needs to exist for disjoint communities. So,

$$t\ (join\ (u_q, nc_i)) \leqslant t\ (join\ (u_q, nc_j)),\ \text{iff}\ (nc_i \cap nc_j = \varnothing) \tag{6}$$

III. Reference Architecture

Figure 2 illustrates how our conceptual model can be subsequently viewed at the reference architecture level. The model shows two communities A and B, with an external Identity Provider (IdP). Each of these three supply members with a contextual identity, shown in the figure as colour-coded keys. Each user, represented by a hexagonal symbol, may contain credentials uniquely identifying the provider. Certain users may contain multiple credentials from different providers, as illustrated by a user simultaneously belonging to Communities A and B with different credentials. As the user has the option of hosting several identities, it can choose to which facet of its identity to present to which community. However, communities can also federate their user identities among themselves, if they choose to trust each other.

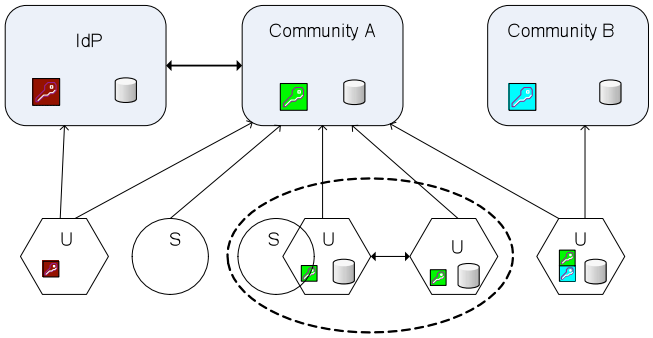


Figure 2. Reference Architecture

Each community also possesses a logical identity manager which uniquely names each community as well as its properties. In addition, communities contain a Service Repository (explained in the following subsection). The reference architecture does not dictate a concrete network space because we envision members to straddle various kinds of access as well as social networks. For example, in

Figure 2, Community A's members can be perceived to arrive from access networks using a provisioning platform such as IMS, or various ISP and home networks, community networks such as FON [1], federated networks which are eduroam™ [2] or OpenID [3] -based, as well as social networks such Facebook. These various networks can either serve individually or in tandem, as implementation platforms.

At the same time, any middleware solution implementing the components of our reference architecture must be able to manage the community policies to take into account the authentication credentials that will be deemed acceptable for service access into that community together with the kinds of Identity Providers deemed suitable. Community services, as well as identity management are discussed in the following subsections.

A. Community services

We take a broad interpretation of community services. A community can offer several kinds of services:

- Connectivity services, which might include providing full or restricted wireless narrowband or broadband network access to a community's users. Connectivity can be infrastructure oriented such as with WiFi access points or it may be infrastructure-less such as with ad-hoc or mesh networks with one or more users acting as gateways. Concrete examples are public WiFi hotspots, as well as community networks such as FON.
- Network services, which aid a member of a community with additional features. These may include enhancing basic connectivity with additional functionality such as native or transitional IPv6 provisioning, tunnel brokering, Virtual Private Networks and Dynamic DNS. They could also encompass infrastructure resources such as access for printing, file storage, email and web servers or even Identity Provision.
- Social and interactive services, which aim at providing users with the ability to collaborate using various methods such as messaging, audio/visual communications, web-based participatory platforms for viewing and publishing blogs and multimedia content as well as social networking.

A community can therefore provide service enablers: small building blocks which the users can use to create a richer set of services of their own. Community services need not reside in one central space, but can be distributed throughout the network, ranging from a user's own terminal or access point, or in some external content provider's network. Examples include providers such as YouTube, Flickr, Google Maps or Apple's AppStore. However, each community needs to have a Service Repository which contains metadata outlining the service portfolio. This can be centrally managed by a logical entity or have a distributed approach such as with Distributed Hash Tables (DHT). In either case, the repository must be capable of defining fine grained access policies per service based on the user credentials within this community. In addition, the repository should also be able to indicate the usage duration for each service as a function of each user's credentials, pricing, tracking usage for accounting and billing purposes. For instance, a user might get authorisation to use the service just for ten minutes. Community policy can also have the possibility of member endorsement, i.e., access to the service is granted provided a suitable amount of members choose to trust the service requestor. In addition, services should provide means to be verified as authentic to users.

B. Community Identity Management

In this sub-section, we use the terminology "entity" to mean a user as well as the community. When it comes to managing identities and their properties, we place following requirements for our architecture.

1) Secure naming of each entity and multiple facets of identity. Every entity has a name, so that it can be pointed to. The entity is also able to provide proof of possession of such a name. An entity can also have several names, thus having a multifaceted identity, and it can choose which facet of its identity to show to which party. An identity is represented with a public key pair and a usable handle is provided by a hashed representation, much in the fashion of ORCHIDs [4].

2) Web of trust and hierarchical trust models. Both the web of trust and hierarchical trust models can be employed. Communities can act as trust anchors, providing hierarchical trust for their members. Members can have direct trust relations with other members and based on these relations, they are able to make assertions. Essentially, when communities negotiate with each other, they create webs of trust. This can take advantage of both the PGP and third party certificate based approaches. However, in terms of certificates, we suggest light weight approaches, such as those provided by SPKI, which also allows identification of entity attributes.

3) Endorsements and delegation of rights. An entity can be in possession of rights, such as a privilege to execute a certain action. Such rights can be delegated to third parties. Thus, it is possible to endorse third parties and provide them authorisations. Privileges and delegations are bound to a secure name. This can take place with SPKI certificates, like suggested above. In terms of users, certain member endorsements can be more valuable, such as those given by members who contribute to the service portfolio of the community.

4) Binding between layers. Some entity names have context in different layers. For instance, an entity could have an application and network level identities. It is possible to make a binding (or delegation) between these two. One example of this is the adoption of Host Identity Protocol (HIP) for the network entities and binding it to the application level interaction, i.e., assurance of network level identity provides increased assurance for the application level identity.

5) Dynamic negotiation of trust relationships. On top of static agreements, entities can negotiate establishment of trust among themselves in a dynamic fashion. For instance, Community A and B can agree that they will provide services to both of their members. Negotiation results in a

policy according to which they will act. This borrows from the ideas presented within the work related to dynamic roaming agreements on the network level [5].

6) Privacy. The user should possess the possibility to enjoy privacy protection. This can take with the help of short term identities, i.e., community assurance is provided only for limited lifetime.

7) Identity provisioning flexibility. Entities have the possibility of choosing their own identity providers. However, when users become members of community, they implicitly submit themselves to the identity provision guidelines of the said community. When a user wishes to use a service, it can provide the identity and identity provider it wishes to use for that transaction. It is up to the policies of the service and its community whether such identity is acceptable. Thus, both models of Liberty Alliance and OpenID are embraced.

8) Mutual authentication of identities. When entities interact, they will both provide their identity and the necessary proof of possession. Thus, there is always certainty that the transaction takes place with the same entity, unless explicitly delegated to a third party. This is the sameness quality of the identity that can be used to establish opportunistic trust.

9) Usability. Users of the system should have a possibility to have meaningful handles to the identities. In other words, when they interact with services, they can employ human readable names. However, those are not used as basis of authentication decisions. The middleware used should internally resolve them to applicable identifiers. Web 2.0 provides examples of creating user interfaces that can hide the complexities of communication from the user.

IV. Implementation And Interaction

We aim at our reference architecture being refined into several kinds of platforms, both existing and for the future. With regard to IP Multimedia Subsystem (IMS) serving as a possible implementation platform, one such community management framework has been discussed in [6] with emphasis on data model. In our model the IMS architecture could provide the signalling plane for the service provisioning, as shown in Figure 3. IMS can provide the centralised version of the community architecture as the identity management while repositories can be handled by centralised entities. For instance, the Home Subscriber System (HSS) can host various member identities. IMS also takes care of the routing of service discovery requests, so the services need to be centrally registered within the system. They also have identifiers on two different levels, because, for instance, SIP URI is used for routing the requests, whereas proof of possession is provided with other identifier, such as hash calculated from the public key. An approach employing HIP and SPKI is presented in [7] and could be adapted to this architecture.

When a member wishes to interact with a community service, there are several options for the verification of the identity. While using networking level technologies like HIP to establish cross layered identity association has its benefits, the first steps of deployment might require less drastic changes. In a simple case, the community has issued a membership certificate that can be presented to the service. This could take place within a modified TLS negotiation or in case higher privacy protection is required, the exchange can take place within the TLS tunnel, which is created using a temporary, short term identity. This could be implemented using AJAX technologies. Another option in terms of identity verification is to involve the identity providing entity (be it IdP or community identity manager) into the online transaction. In this case IdP is contacted either directly by the service or via the client and an authorisation to use the service is requested.

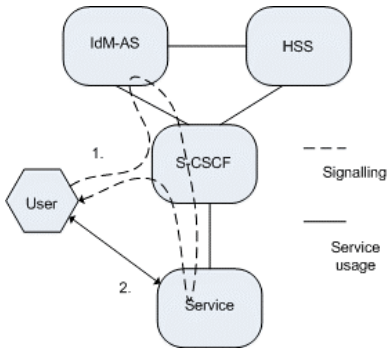


Figure 3. Implementation Example Using IMS

V. Example Scenarios

A. Community networking

Wireless Tampere [8] is a city-wide effort to provide Wireless LAN access in the city of Tampere, Finland. In stark contrast to the centralised infrastructure and management of wireless hotspots by many other cities worldwide, Wireless Tampere is a community-driven broadband access network. Participating members and organizations offer some surplus bandwidth in their WiFi networks to the community. In return, they are able to obtain free WiFi connectivity from other participating users in Tampere. A separate authority, the coordinator, monitors and controls the community network brand and defines the set of rules and interfaces as to how the service and network solution providers interact with the community and each other. Authentication is based on hierarchical RADIUS federations among member organizations, and both WPA as well as WWW-based authentication is supported for users. Non-members are allowed time-limited access to this community network for a fee.

The scenario that applies the suggested concepts involves the creation of a new online community that actively contributes content for the city for purposes of tourism or social interaction, as opposed to a static website used by the city's tourism board. Active members provide multimedia content such photos, recorded or live video streaming around the city for various purposes, ranging from webcams to covering specific events like festivals and parades. This could also include news, “as-they-happen”, kind of microblogging. Live multimedia content can be considered a

value added service, offered as a paid service to tourists or Tampere residents temporarily residing elsewhere. The community can also find some way to reward and engage tourists interested in offering content. Examples of rewards include tokens for use in the city's services such as extended or higher speed connectivity, parking, bus fares as well as food and beverages from some outlets.

This scenario opens up the several implementation possibilities that we aim to encompass with our architecture. Firstly user created content becomes part of the community service portfolio and additionally a web portal could provide proxying in order to provide better performance. The community could also grant an authorisation certificate in order to acknowledge the right to do such content streaming. Proxying could be seen as an implicit permission as well. In order to access the stream, other users ought to present their membership certificates. Streaming authorisation would be presented during the same handshake step. This handshake could take place with the use of HIP like procedure, which allows exchanging the identities and proving secure association between the peers. Thus, origin of the stream can be assured and also assured accounting records can be created.

One possible implementation platform could be IMS (to emphasise the multimedia nature of the service), but other kind of settings are possible as well. The community could also select not to use membership certificates, but instead provided the online authentication service itself to the members wishing to ascertain the membership status of others, such as using an OpenID-oriented approach. Temporary members and tourists may also elect to be identified based on their social network credentials. Thus, the community has a good position of functioning as an identity exchange point. These application level identities can be used to enhance the trust levels on network level in various key exchange protocols, not just HIP. In other words, reputation in community context is transferred (or delegated) to trustworthiness in another context. This can be especially beneficial if the community reputation is based on similarity or friends' ratings [9]. [10] presents one such system that implements IPsec tunnelling using Facebook authentication and list of friends.

B. Extended Homes

Home networks possess certain characteristics that set them apart from other kinds of access and core networks. One of the defining characteristics of the home network is that its nature within the home is strongly shaped by the profile of one or a few family members. A typical home network owner, though possibly technically adept, is more than likely a user who does not wish to engage very heavily in time consuming tasks such as handling the intricacies of installing, maintaining and administering a home network.

Home networks also tend to temporally evolve as a function of the dynamics of the physical home as well as the family members using the home network. The original home network may become divided into two or more disparate home networks which are geographically separated. However, these may need to be occasionally interconnected temporarily to become a virtual extended home network. On the other hand, considering advances in vehicular as well as personal area networks today, it is conceivable that within a short space of time (such as a single day) the devices and services in a single home network may change, as a result of the proximity of family cars or family members moving in and out of the house whilst carrying several portable network devices.

In an extended home scenario one can envisage a simple case where a mobile family member or a relative wishes to access some media or the services provided within the home network. This could be as simple as sharing some holiday photos. It could also involve more complexity such as several people situated within and outside the home, watching a live or recorded IPTV program while using an instant messaging platform to communicate and comment. Mobile members could possess member certificates, which allow them first to request service from the home gateway, so that appropriate pinholes can be punched into the firewall. Another option is that they carry personal endorsements given to them by the other family members, for instance, during a prior family event. Endorsements can be transferred or exchanged between the devices with the help of near-field communication such as Bluetooth, or allowing tapping devices against each other. Naturally, the identity of the mobile member can have been preconfigured as trusted one by typing in the hashed representation of it to the management system. Thus, only proof of possession is required. Note that even though "first level service" is given by the firewall, the actual service, such as sharing of pictures, still verifies the identity.

However, a technically adept home owner could also use his residential gateway to provide network level services to other mobile family members, relatives or friends. For example, [11] discusses how the home gateway acts as an IPv6 border router, delivering IPv6 addresses to other devices outside the home, using a VPN connection. [12] introduces a new flexible sharing model built atop such an IPv6 overlay network, which allows family members and friends to obtain as well as offer content distributed in various devices in a decentralised manner regardless of location. Instead of having the home VPN server generate certificates for distribution to connecting devices, a more dynamic approach would be to employ the connecting user's social network credentials to establish the VPN connection, as discussed in [10].

VI. FUTURE RESEARCH CHALLENGES

While we envisage an architecture for trusted communities, it has to be kept in mind that trust is a challenging topic, especially when it comes to automated negotiation. Liability can become an issue and the amount of doubt increases with deeper hierarchies. However, combining both hierarchical and web of trust approaches, along with assured identities, provide means to investigate reputation based systems that can converge quicker to higher trust values. This also has to factor the bootstrapping of trust and total value of individual endorsements, so that Sybil-like attacks employing multiple fake identities are not a concern.

This paper presents an outline of our architecture and needs to be augmented with a proper implementation to study its feasibility. The Web 2.0 world provides mechanisms for interacting with the user, but user interface issues as well as correctly resolving user readable identifiers need to be overcome. In the face of phishing threats, a password is something that is easy to use but easier to lose, while a public key based approach, for instance, can be quite the contrary.

It can be argued that the performance of such a public key based community identity system might be inadequate, or the possibility of HIP support is unrealistic deployment-wise. However we expect gradual introduction of the various parts of the architecture in selected communities while additional steps can be undertaken when certain technologies are accepted widely enough. We believe the amount of dynamic interaction for users as well as for service composition and usage, will vastly increase in future networking scenarios involving ambient intelligence and the like. Thus, solutions that give more assurance to the identities and the service transactions are crucial.

REFERENCES

[1] Fon WiFi Community, http://www.fon.com

[2] K. Wierenga, L. Florio, “Eduroam, past, present and future”, Proceedings of the TERENA Networking Conference 2005, Jun 2005.

[3] D. Recordon, D. Reed, “OpenID 2.0: a platform for user-centric identity management”, Proceedings of 2nd ACM Workshop on Digital Identity Management, Nov 2006.

[4] P. Nikander, J. Laganier, F. Dupont, "An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers(ORCHID)", IETF RFC 4843, Apr 2007.

[5] 3GPP. Network Composition Feasibility Study. 3rd Generation Partnership Project Technical Report. TR22.980 V8.1.0. June 2007.

[6] A. Yamamoto, Y. Araki, M. Sweeney, "A Framework of Community Management with Object Deputy Mechanism for IP Multimedia Subsystem", in Proceedings of the 2008 International Symposium on Applications and the Internet, Jul 2008.

[7] S. Heikkinen, "Establishing a Secure Peer Identity Association Using IMS Architecture", Proceedings of The Third International Conference on Internet Monitoring and Protection, Jul 2008

[8] K. Huhtanen, H. Vatiainen, S. Keski-Kasari, J. Harju, “Utilising eduroamTM architecture in building wireless community networks”, Journal of Campus-Wide Information Systems 2008 Vol 25 No 5 pp 382-391.

[9] C. Jensen, J. Davis, S. Farnham,, ”Finding Others Online: Reputation Systems for Social Online Spaces”, Proceedings of the 2008 SIGCHI Conference on HumanFactors in Computing Systems, Apr 2002.

[10] R.J. Figueiredo, P. O. Boykin, P. St. Juste, D. Wolinsky, “Social VPNs: Integrating Overlay and Social Networks for SeamlessP2P Networking”, Proceedings of the 2008 IEEE 17th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Jun 2008.

[11] K. Huhtanen, B. Silverajan, J. Harju, “Utilising IPv6 over VPN to Enhance Home Service Connectivity,” Terena 2007 Special Issue of the Journal of Campus-Wide Information Systems Vol 24 No 4 pp 271-279.

[12] B. Silverajan, A. Vekkeli, T. Vartiainen, J. Harju, “Facilitating Content Exchange Among Homes, Ad-Hoc Communities and Mobile Users,” Proceedings of 13th IEEE International Symposium on Consumer Electronics (ISCE 2009), May 2009.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-2684-8
ISSN 1459-2045