Project Number:	**Contract Number: INFSO-ICT-*224287***
Project acronym:	**VITAL++**
Project Title:	**Embedding P2P Technology in Next Generation Networks: A New Communication Paradigm & Experimentation Infrastructure**
Title of Report	**Authentication and Accounting**

Instrument:	STREP
Theme:	ICT-2-1.6
Report Due:	M21 (Feb.2010)
Report Delivered:	
Lead Contractor for this deliverable:	UoP
Contributors to this deliverable:	Shane Dempsey (WIT), Kabir Ahmed (WIT), Stephen Garvey (WIT), Jonathan Brazil (WIT), Odysseas Koufopavlou (UoP), Charalabos Skianis (UoP), Apostolos P. Fournaris (UoP), Nikolaos Efthymiopoulos (UoP), George Karidis (BCT), Jose Luis Pena (TID),Juana Sanchez (TID).
Estimated person months:	15
Start date of project:	1st June 2008
Project duration	30 months
Revision:	Version 1.0
Dissemination Level:	Public
Internal reviewer:	Wolfgang Brandstätter (TA)

This page has been left blank intentionally.

1 Table of Contents

2 List of Figures

3 Executive summary

This deliverable describes the state of the art in Authentication and Identity management for P2P and IMS. The state of the art in IMS and Internet accounting technologies is also presented. The Vital++ solutions for authentication in a P2P-IMS environment are specified. These solutions include a Public Key Infrastructure system for peer authentication, which is reused for accounting and content licensing purposes. Moreover, the functionalities and operations of the Vital++ Accounting Subsystem are described.

This document is complementary to D4.2, which is about the SoftMix service scenario representing a specific use case of the authentication and accounting solutions described within D4.1. It is also complementary to D4.3, which describes the Content Protection Subsystem that interacts with the Accounting subsystem during the licensing process.

4 Introduction

The purpose of this document is to review the state of the art in two related areas, which are important for the successful realisation of the Vital++ platform; Identity Management and AAA.

By Identity Management we refer to the management of user credentials. We also refer to the association of different subscriber identities used for access to multiple services or service domains with a single logical identity corresponding to a user. From an IMS viewpoint this single user is a subscriber to a network operator and the process whereby user addresses or identifiers at multiple domains are mapped to the single billable identifier is known as “identity convergence”. Systems like Vital++’s proposed P2P-IMS platform challenge this identity model by introducing interchangeability of service consumer and provider roles, in the case of user generated content, and also promoting interaction with for-profit content networks with their own concepts and specifications for identity.

By Authentication we refer to the mechanisms whereby a user or service provider can prove their identity to a network operator or 3rd party service and content provider. The 3GPP have described native mechanisms for authentication within the IMS specifications. These have been augmented by ETSI TiSPAN (Telecommunications & Internet Converged Services and Protocols for Advanced Networking). The TISPAN standards track deals with legacy system interworking issues that are prevalent in fixed-IMS deployments. TISPAN indicates approaches to interworking an IMS system with a legacy or IMS-unaware access network. These are further elaborated in 3GPP’s Generic Authentication Architecture (GAA) standards track. Consideration of these specifications will enable the Vital++ project to indicate a practical approach to the challenges of Identity Management & Authentication within a P2P network.

Identity in the context of a P2P overlay poses its own unique challenges as network authentication may be decentralised. The Vital++ project describes a P2P IMS environment where strong IMS authentication is available to Vital++ subscribers.

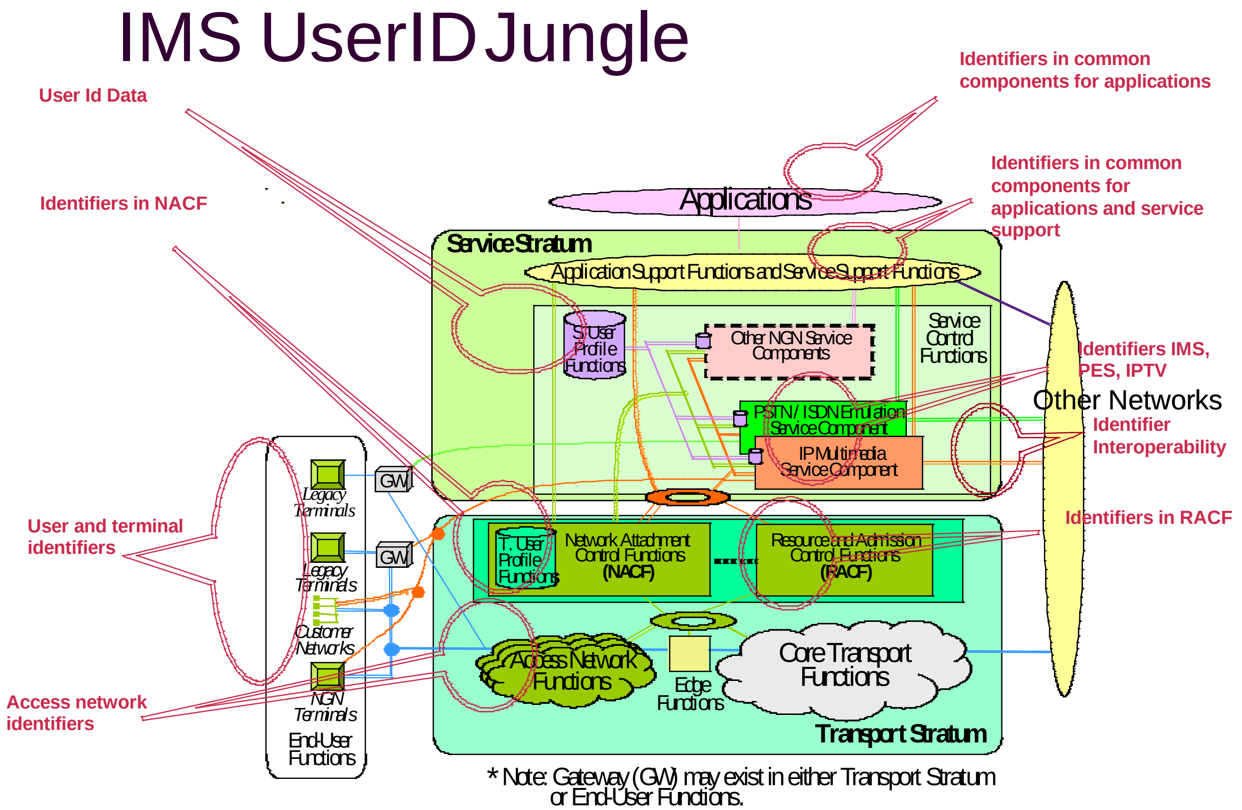
Accounting in Vital++ attempts to blend IMS accounting specifications for pre-pay and post-pay with P2P accounting mechanisms to incentivise good network behaviour. However, it is not intended to replace an operator’s existing charging and billing infrastructure. Instead it develops a content-charging model where charging data, rating schemes and resulting subscriber bills are generated in the context of content licensing. The CREST system described within this deliverable is triggered by the Content Protection Subsystem described within D4.1.

The Accounting Subsystem provides interfaces for Content Providers to associate a charging scheme with their content. It also exposes web service and diameter interfaces to receive network and content based charging information. The actual rating is carried out using the Internet Protocol Detail Records. Rating schemes are spreadsheet worksheets which can support arbitrarily complex rating schemes incorporating conditional rules.

5 State of the Art in IMS Identity & Authentication

In this chapter, we describe the state of the art in technologies and specifications for Identity Management and Authentication of IP Multimedia Subsystems defined by 3GPP and ETSI. By authentication we primarily refer to mechanisms for subscriber authentication for the purposes of getting access to network provided services (e.g. voice and messaging). We'll also consider, however, mechanisms for service provider authentication. This is necessary because Vital++ users can provide content distribution services based on the Vital++ P2P-IMS platform using their peer software. Therefore, we must understand how Vital++ peers can utilise IMS AAA mechanisms effectively given their potential dual role.

5.1 Summary of IMS & NGN identity challenges



IMS UserID Jungle
User Id Data
Identifiers in NACF
Identifiers in common components for applications
Identifiers in common components for applications and service support
Identifiers IMS, PES, IPTV
Identifier Interoperability
Identifiers in RACF
User and terminal identifiers
Access network identifiers
Applications
Service Stratum
Application Support Functions and Service Support Functions
S. User Profile Functions
Other NGN Service Components
Service Control Functions
PSTN / ISDN Emulation Service Component
IP Multimedia Service Component
Other Networks
GW
GW
Legacy Terminals
Legacy Terminals
Customer Networks
NGN Terminals
End-User Functions
T. User Profile Functions
Network Attachment Control Functions (NACF)
Resource and Admission Control Functions (RACF)
Access Network Functions
Edge Functions
Core Transport Functions
Transport Stratum
* Note: Gateway (GW) may exist in either Transport Stratum or End-User Functions.

Figure 1 - Overview IMS Identity Formats

IMS is a technology for Next Generation Networks meaning that signalling and media messages are exchanged by the use of the Internet Protocol within the IMS core network. IMS has been designed from the start to integrate with a Network Operator's legacy systems. Hence, it has various identifiers defined throughout the IMS and NGN architecture. IMS identifiers are standalone, isolated within component/stratum making it difficult to correlate IMS identifiers across strata/layers

Strong identities are prerequisite for secure and trustworthy e-business in third and next generation networks. Strong identity means that the subscriber must authenticate using an appropriate scheme to verify their identity and to permit the network operator to confidently make service provisioning, delivery and charging decisions based on that identity.

Every IMS architecture needs to leverage such identities for the purpose of

- Secure identification and authentication (user/device),
- Assisting towards establishing secure communications,
- Protection of the network infrastructure.

The Generic Authentication Architecture (GAA)[1] specification described later on in this deliverable is an attempt to integrate the various identity standards and specifications existing in every IMS network.

5.2 IMS Authentication Model

Authentication between the subscriber and the network shall be performed in a way to achieve mutual authentication between IMS Identity Module (included in the UE - ISIM) and the home network.

The subscriber profile will be located in the HSS of the home network. Any kind of information, considered important for the home network, will be included in the HSS. This information may not be disclosed to an external partner.

At the registration procedure, an S-CSCF is assigned to the subscriber by the I-CSCF. The subscriber profile will be downloaded to the S-CSCF from the HSS. When a subscriber requests access to the IMS Core Network the assigned S-CSCF will check, by comparing the request with the subscriber profile, if the subscriber is allowed to continue with the request or not [i.e. Home Control (Authorization of IM-services). For IMS services, a new security association is required between the UE and the IMS before access is granted to IM-services]. To summarise, the Home Network must always authenticate the subscriber via the registration or re-registration procedures.

[1] 3GPP TS33.220 Generic Authentication Architecture; Generic Bootstrapping Architecture, http://www.3gpp.org/FTP/Specs/html-info/33220.htm

The mechanism for authentication and key agreement is called IMS AKA (authentication and key agreement). This concept is reused in a similar way for UMTS schemes as it provides for secure communications. The identity used for authenticating a subscriber is the private identity, IMPI (IM private identity). The HSS and the ISIM share a long-term key associated with the IMPI.

The home network shall decide the right scheme to use, depending on the access network that the subscriber is using. All parameters (i.e. keys) that are required for this mechanism are transported by the use of SIP.

- **Authentication signalling flow**

If an IM-subscriber is trying to get access to an IM service, at least one public identity (IMPU) must be registered and its private identity (IMPI) must be authenticated at application level. In order to get registered, the UE sends a SIP REGISTER message towards the S-CSCF (SIP registrar server), which will perform the authentication of the user. The message flows are the same regardless of whether the user has an IMPU already registered or not.

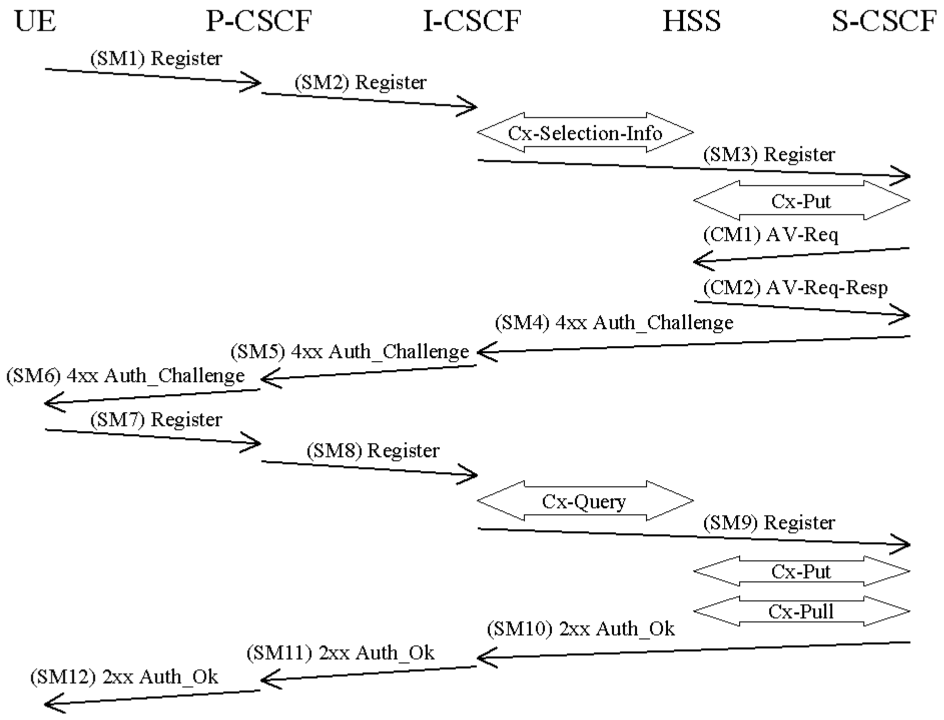


Figure 2 - IMS mutual authentication procedure (according to 3GPP)

NOTE: SMn stands for SIP Message n and CMm stands for Cx message m.

The signalling flow of the figure represents the following messages:

SM1:
REGISTER(IMPI, IMPU)

The SIP messages SM2 and SM3 just represent the forwarding process to the S-CSCF.

In the case of an IMPU not registered at the S-CSCF, after receiving the forwarded message, the S-CSCF needs to set the registration flag at the HSS to initial registration pending. This is done in order to handle UE terminated calls while the initial registration is in progress and not successfully completed. If the IMPU is currently registered, the S-CSCF shall leave the registration flag set to *registered*. At this stage the HSS has performed a check that the IMPI and the IMPU belong to the same user.

Upon receiving the SIP REGISTER the S-CSCF shall use an Authentication Vector (AV) for authenticating and agreeing a key with the user. If the S-CSCF has no valid AV then the S-CSCF shall send a request for AV(s) to the HSS in CM1 together with the number m of AVs wanted where m is at least one.

CM1:
Cx-AV-Req (IMPI, m)

Upon receipt of a request from the S-CSCF, the HSS sends an ordered array of *n* authentication vectors to the S-CSCF using CM2. The authentication vectors are ordered based on sequence number. Each authentication vector consists of the following components: a random number RAND, an expected response XRES, a cipher key CK, an integrity key IK and an authentication token AUTN. Each authentication vector is valid for just one authentication and key agreement between the S-CSCF and the IMS user.

CM2:
Cx-AV-Req-Resp (IMPI, RAND1-AUTN1-XRES1-CK1-IK1,….,RANDn-AUTNn-XRESn-CKn-IKn)

When the S-CSCF needs to send an authentication challenge to the user, it selects the next authentication vector from the ordered array, i.e. authentication vectors in a particular S-CSCF are used on a first-in / first-out basis.

The S-CSCF sends a SIP 4xx Auth_Challenge i.e. an authentication challenge towards the UE including the challenge RAND, the authentication token AUTN in SM4. It also includes the integrity key IK and the cipher key CK for the P-CSCF. The S-CSCF also stores the RAND sent to the UE for use in case of a synchronization failure.

SM4:
4xx Auth_Challenge (IMPI, RAND, AUTN, IK, CK)

When the P-CSCF receives SM5 it shall store the key(s) and remove that information and forward the rest of the message to the UE.

SM6:
4xx Auth_Challenge (IMPI, RAND, AUTN)

Upon receiving the challenge, SM6, the UE takes the AUTN, which includes a MAC and the SQN. The UE calculates the XMAC and checks that XMAC=MAC and that the SQN is in the correct range. If both these checks are successful the UE uses RES and some other parameters to calculate an authentication response. This response is put into the Authorization header and sent back to the SIP registrar server specifies how to populate the parameters of the response. It should be noted that the UE at this stage also computes the session keys CK and IK.

SM7:
REGISTER (IMPI, Authentication response)

The P-CSCF forwards the authentication response in SM8 to the I-CSCF, which queries the HSS to find the address of the S-CSCF. In SM9 the I-CSCF forwards the authentication response to the S-CSCF.

Upon receiving SM9 containing the response, the S-CSCF retrieves the active XRES for that user and uses this to check the authentication response sent by the UE. If the check is successful then the user has been authenticated and the IMPU is registered in the S-CSCF. If the IMPU is not registered, the S-CSCF shall send a Cx-Put to update the registration-flag to *registered*. If the IMPU is registered the registration-flag will not be altered.

5.2.1 AAA for Service Providers

The IP Multimedia Subsystem (IMS) at its core has a design principle, which determines its approach to security. IMS keeps the signalling and media paths separate, Network nodes do not have to handle both. Since the sixth release of the 3rd Generation Partnership Project (3GPP) IMS provides support for different access networks. This access independence introduces multiple access requirements. In order to meet these requirements IMS reuses protocols developed by the European Telecommunications Standards Institute (ETSI) for Global System for Mobile (GSM) networks as well as access protocols developed by the Internet Engineering Task Force (IETF). Authentication, Authorisation, and Accounting (AAA) is achieved in IMS through the use of the IETF Diameter protocol.

The Session Initiation Protocol (SIP) has been chosen by the 3GPP and Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN) as the session control protocol in IMS. IMS reuses a concept from GPRS and GSM networks of a home and a visited network. An IMS user is allocated one or more Public User Identities (IMPU) by the home operator and this is used to route SIP signalling. The user is also allocated a Private User Identity (IMPI) which is used for subscription identification and authentication. The IMPU is equivalent to the MSISDN in GSM networks and the IMPI is equivalent to the International Mobile Subscriber Identity (IMSI) which is stored in the Subscriber Identity Module (SIM) card.

The IMS architecture can be split into three horizontal layers; the Service Layer, the Control Layer, and the Connectivity Layer. The service layer comprises application and content servers. The control layer comprises network control servers which manage the call or session. The connectivity layer comprises routers and switches.

The outcome is an overall architectural framework and suite of protocols which provide the necessary solutions to enable inter-operator roaming, QoS and reliable user-friendly charging.

5.2.2 AAA and Diameter

By providing an ID and password a user can gain access to a server, which in turn offers a variety of services based on the user credentials. Generally, the user's credential information is not stored directly on the access server, but some other secure location such as a Lightweight Directory Access Protocol (LDAP) server behind a boundary firewall. Therefore, a standardised protocol is required between the access server and the user information repository in order to exchange *authentication-, authorization-, and accounting*-related information. The RADIUS protocol was designed to provide a simple, but efficient, way to deliver such AAA capability.

With the evolution of networks applications and protocols, new requirements and mechanisms are implemented to authenticate users. These requirements include topics like failover, security, and audit ability. Although there are some subsidiary protocols intended to extend the capability of the RADIUS protocol, a more extensible and general protocol was required. The Diameter protocol was derived from RADIUS, and designed to be a general framework for future AAA applications.

Diameter includes numerous enhancements over RADIUS, such as error handling and message delivery reliability. It extracts the essence of the AAA protocol from RADIUS and defines a set of messages that are general enough to be the core of the Diameter Base protocol. The applications which require AAA functions can define their own extensions on top of the Diameter base protocol.

Diameter operates on top of reliable transport protocols like TCP and SCTP. The Base Diameter Protocol provides the following basic services:

- Delivery of AVPs
- Capability Negotiation
- Error Notification
- Accounting
- Extensibility via new command codes and AVPs

5.2.3 IMS Identity Management Using GAA

As mentioned in the previous section users are required to have credentials for each service they wish to access. Either they have to enter usernames and passwords when challenged or the credentials are preconfigured on the UE or client application. The existence of several sets of credentials is not only an inconvenience for the user but it is also expensive for the operator and service providers to provision.

The Generic Authentication Architecture (GAA) is a solution for the growing need for authentication and key agreement between the client and the services on the Internet or in the mobile operator's network. GAA tackles this problem by using the existing GSM authentication system as a basis for providing new credentials for both clients and servers. The mobile network operator provides an authentication service using GAA which lets the client and the service authenticate each other by allowing the parties involved to exchange shared secrets utilising the existing 3G Authentication and Key Agreement (AKA) authentication system.

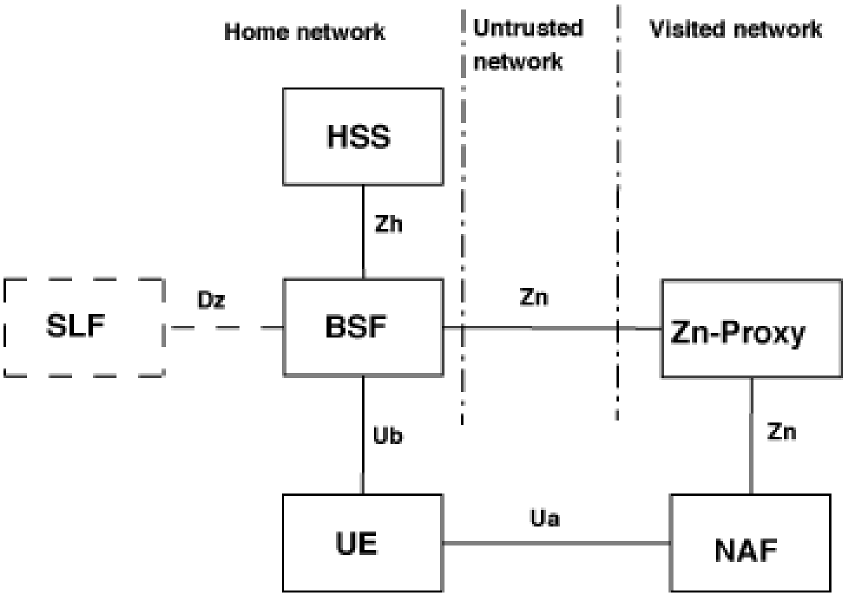


Figure 3 - IMS Nodes involved in GAA "boot-strapping"

The GAA can be used in two ways. The first uses a shared secret between the server and the UE, the second uses public private key pairs and digital certificates. For the shared secret scenario the client and operator are authenticated using 3G AKA. They agree on a set of session keys which the client will use later on to access the services. This procedure is also known as bootstrapping. Once this is done the different services can retrieve session keys from the operator and use them to provide a service between the client and the service provider.

In the second GAA scenario the bootstrapping occurs as mentioned above. However the client then requests certificates from the operators' or service providers' PKI infrastructure. Authentication is achieved by using the session keys created during bootstrapping. These certificates and keys are then used to produce digital certificates or authenticate the client with the server to access services.

The UE and the Bootstrapping Server Function (BSF) mutually authenticate themselves over the Ub interface, by using the HTTP Digest AKA protocol. The UE also communicates with the Network Application Functions (NAF), which are the application servers, over the Ua interface, which can use any application specific protocol necessary.

The BSF retrieves the subscriber's data from the Home Subscriber Server (HSS) over the Zh interface, which uses the Diameter Base Protocol. If there are several HSSs in the network, BSF must first figure out which one to use. This can be done by either configuring a pre-defined HSS to BSF, or by querying the Subscriber Locator Function (SLF) over the Dz interface.

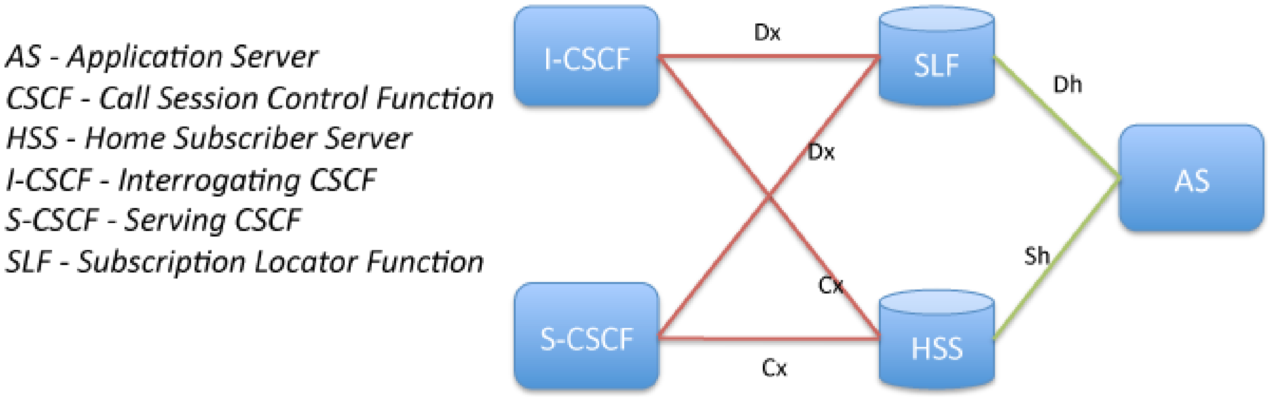

AS - Application Server
CSCF - Call Session Control Function
HSS - Home Subscriber Server
I-CSCF - Interrogating CSCF
S-CSCF - Serving CSCF
SLF - Subscription Locator Function
I-CSCF
Dx
SLF
Dh
Dx
AS
Sh
Cx
S-CSCF
HSS
Cx

Figure 4 - Interfaces for IMS Authentication

NAFs retrieve the session keys from BSF over the Zn interface, which also uses the Diameter Base Protocol. If NFA is not located in the home network, it shall use a Zn-Proxy to communicate with BSF.

Diameter is used on the Sh and Cx interfaces defined by 3GPP for the IMS. The Sh and Cx Diameter applications extend the Base Diameter Command codes and AVPs to support the authentication and authorisation functions required for the respective interfaces. The figure above depicts these interfaces in the IMS network along with the Dh and Dx interfaces.

The Sh interface operates between a SIP AS and the HSS network elements in the IMS. The Sh interface allows for:

- Download and update of transparent and non-transparent user data
- Request and send notifications on changes in the user data

The Dh interface is used between the AS and the SLF. It is used to get the address of the HSS serving an IMS Public User Identity or Public Service Identity. The Dh interface uses the same message set as the Sh interface.

The Cx interface operates between I-CSCF and HSS and between S-CSCF and the HSS. The Cx interface allows for:

- Location management procedures (exchange of location information)
- User data handling procedures (download user data stored in the server)
- User authentication procedures

HSS implements the Diameter Multimedia server side of the Cx interface while the I-CSCF and the S-CSCF implement the Diameter Multimedia client side of the Cx interface.

The Dx interface is used between the Call Session Control Function (CSCF) and the Subscriber Locator Function (SLF). It is used to get the address of the HSS serving an IMS Public User Identity or Public Service Identity. The Dx interface uses the same message set as the Cx interface.

For charging, the 3GPP defines two types of interfaces. The online charging interface (Ro) is used for real-time billing while a service is executed. Charging information can affect the service being rendered. The offline charging interface (Rf) is used to transfer charging information that will not affect, in real-time, the service being rendered.

A Diameter message is the base unit to send a command or deliver a notification to other Diameter nodes. For different purposes, Diameter protocol has defined several types of Diameter messages, which are identified by their command code. For example, an Accounting-Request message recognizes that the message carries accounting-related information, while a Capability-Exchange-Request message recognizes that the message carries capability information of the Diameter node sending the message.

Because the message exchange style of Diameter is synchronous, each message has its corresponding counterpart, which shares the same command code. In both previous examples, the receiver of an Accounting-Request message prepares an Account-Answer message and sends it to the original sender.

The command code is used to identify the intention of a message, but the actual data is carried by a set of Attribute-Value-Pairs (AVPs). The Diameter protocol has predefined a set of common attributes and imposes each attribute with a corresponding semantic. These AVPs carry the detail of AAA as well as routing, security, and capability information between two Diameter nodes. In addition, each AVP is associated with an AVP Data Format, which is defined within the Diameter protocol (for example, OctetString, Integer32), so the value of each attribute must follow the data format.

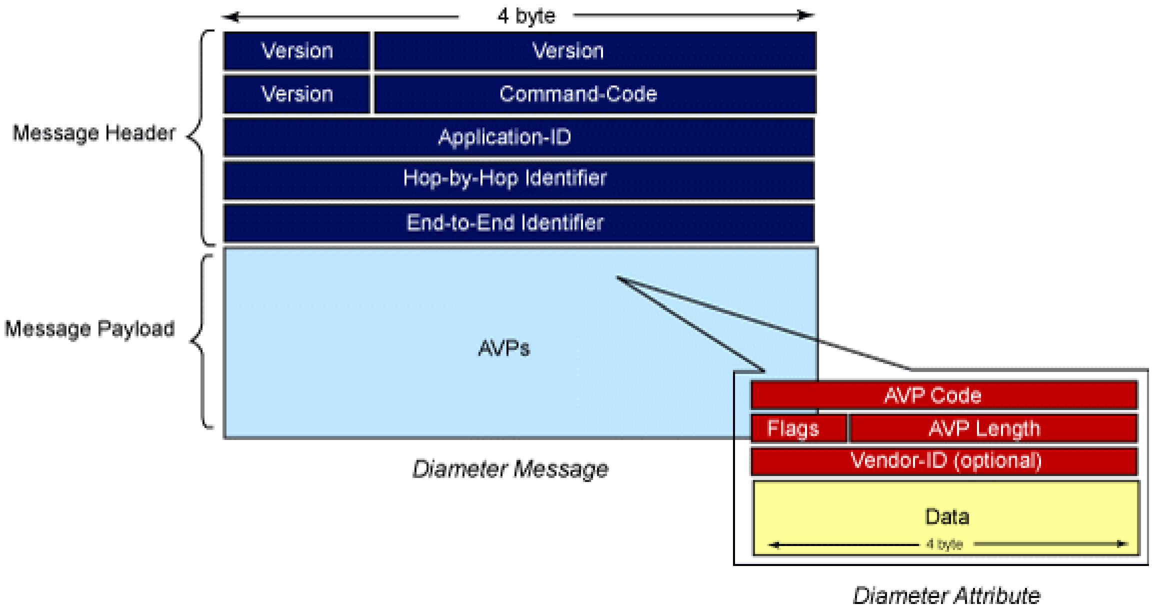


Figure 5 - Diameter Protocol Struture

5.2.4 TISPAN NGN AAA

Initially TISPAN worked on harmonizing the IMS core for both wireless and wireline networks. However in early 2008, the common IMS specifications were transferred back to 3GPP so that one unique standards organization is responsible for providing a Common IMS fitting any network needs (fixed, 3GPP, CDMA2000, etc.). TISPAN has an active interest in providing authentication support to legacy terminals. A legacy terminal in this scenario equates to a PC or other device which does not have a Universal Integrated Circuit Card *(*UICC) which contains parameters used for identifying and authenticating the User Equipment (UE) to the IMS. Therefore TISPAN introduces its own approaches to identify and authenticate users and UE:

- ***NASS Bundled Authentication (NBA):*** Utilises the results of access-layer authentication for the IMS-Layer
- ***IMS Residential Gateway (IRG):*** Acts as a ISIM/UICC adapter between legacy terminals and the IMS core in such a way that IMS-AKA can be used between the IRG and the IMS core network.
- ***Residential Gateway (RGW) or Access Gateway (AGW):*** These are controlled by the Access Gateway Controller Function (AGCF). The AGCF acts as a combination of both a terminal and an outbound SIP proxy towards the IMS core (Proxy-CSCF)

5.2.5 NASS Bundled Authentication (NBA)

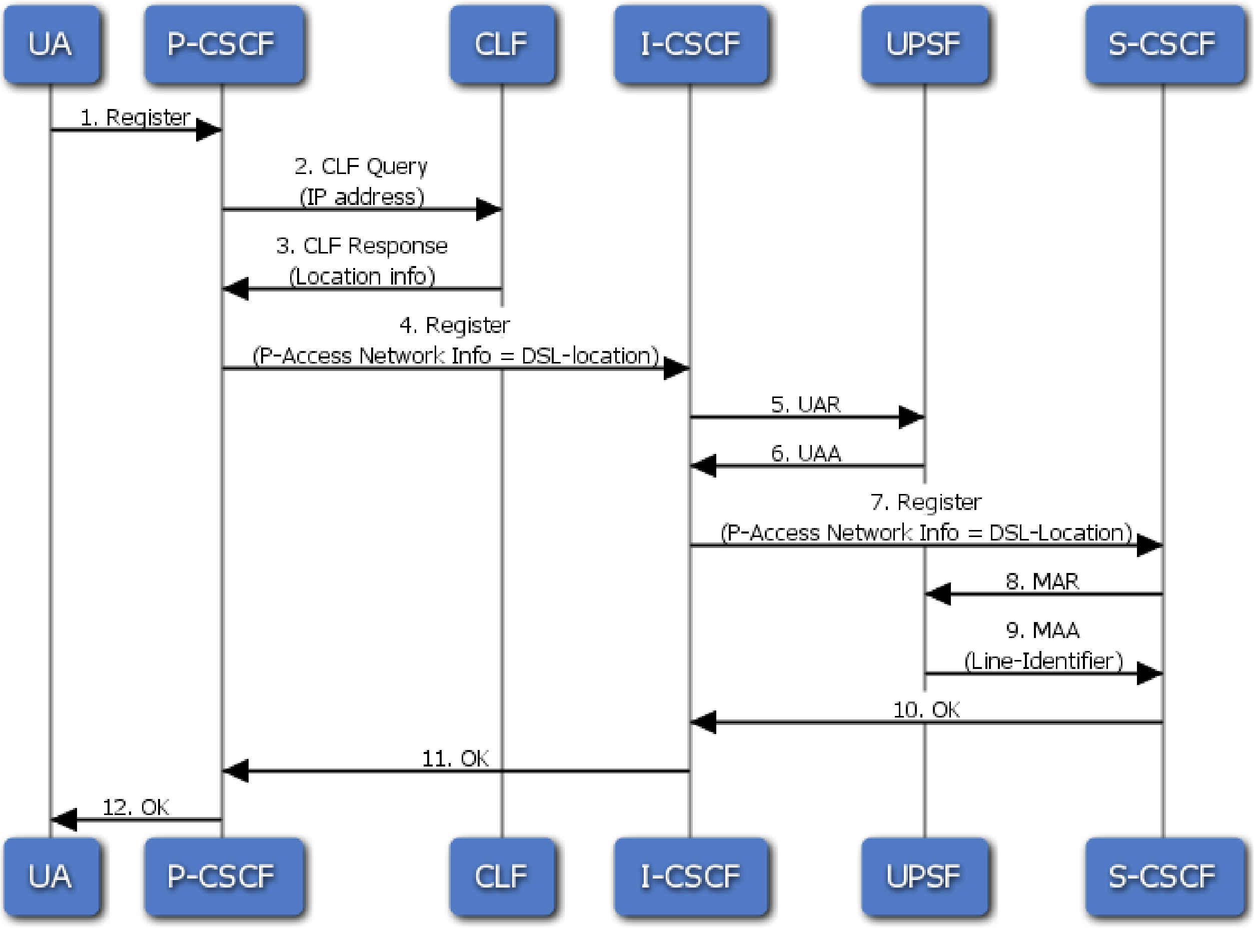


Figure 6 - NASS Bundled Authentication Sequence Diagram

The Network Attachment Subsystem (NASS) Bundled Authentication based solution uses a successful authentication to the access network to obtain IP connectivity. IMS then uses the result of this access level attachment as the input for IMS-level authentication. This approach was used by the 3GPP as an early IMS security solution.

5.2.6 IMS Residential Gateway (IRG)

The IRG is an adapter between the UICC-less terminal and the 3GPP IMS core that requires a ISIM/UICC and therefore provides another approach for legacy terminal support. The IRG is equipped with a ISIM application on a UICC and implements a SIP Back to Back User Agent (B2BUA) providing a Gm interface between the UE and the IMS Network. In the IRG many terminals can be attached but they all share the same IMS identity represented by the ISIM on the UICC. A single IMS identity for several terminals may introduce billing and roaming problems.

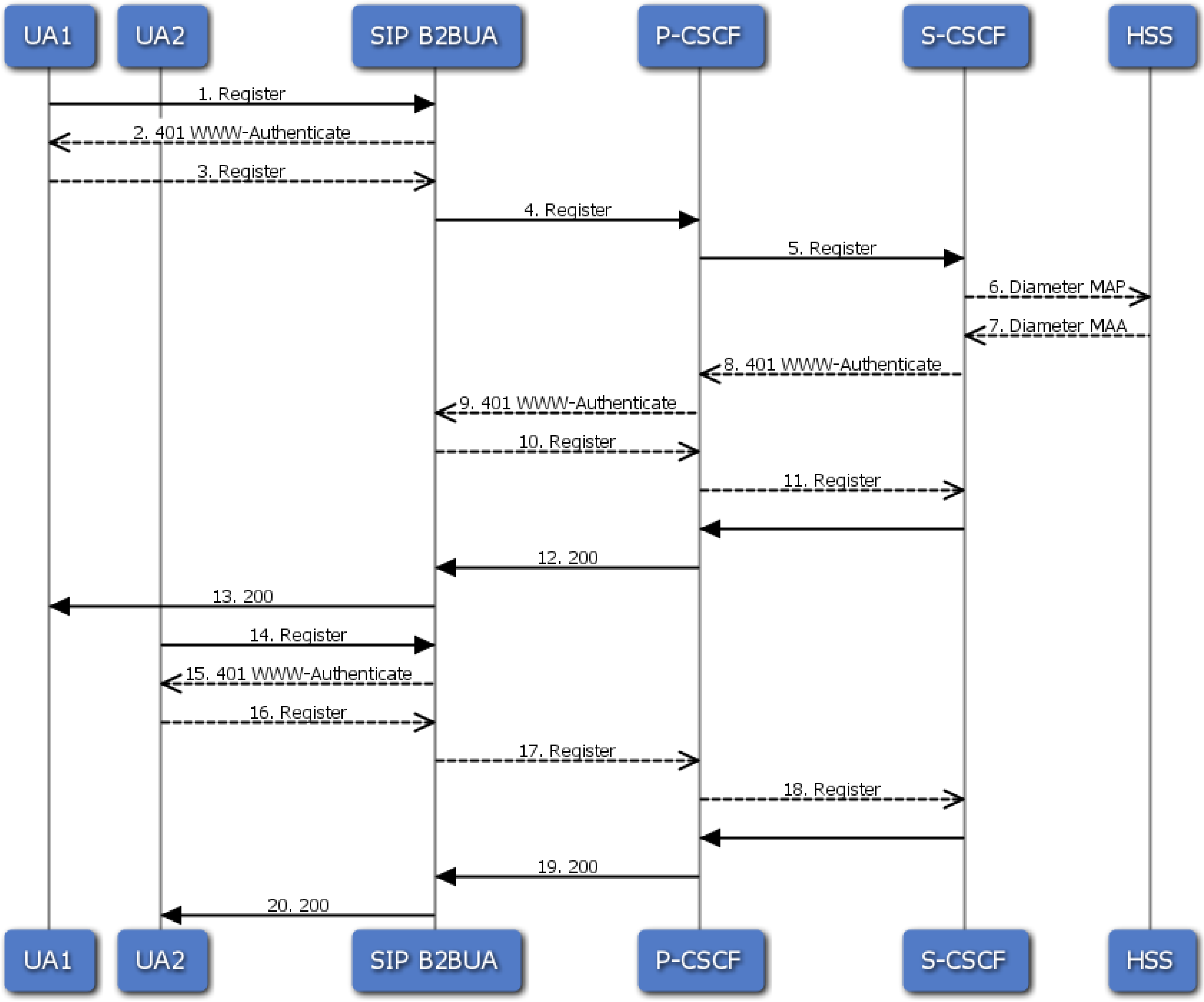


Figure 7 - IMS Residential Gateway using B2BUA

5.2.7 Residential Gateway / Access Gateway

The use of a RG or AG controlled by an AGCF provides another means for legacy terminal equipment to connect to an IMS network. The legacy terminal can connect using a Residential Gateway at the user's location or a centralised Access Gateway located at the operators location. The RGW and the AGW provide similar functionalities but the difference between them is in terms of scale. The RGW serves a couple of terminals while the AGW scales to thousands of terminals. Both the RGW and the AGW are controlled by the

AGCF that represents the UE and the Proxy-CSCF functionality towards the IMS core.

5.2.8 GBA

The Generic Bootstrapping Architecture (GBA) offers a generic authentication capability for various applications based on shared secret. Subscriber authentication in GBA is based on HTTP Digest AKA, RFC 3310 [1]. The GBA model is used to authenticate subscribers wishing to access network multimedia services. (e.g. voice, video calling, etc.) .

Support of subscriber certificates and Access to Network Application Function using HTTPS is based on GBA.

GBA, Subscriber certificates, and Access to Network Application Function using HTTPS Transport Layer Security (TLS), an enhancement to GBA provided by the Generic Authentication Architecture (GAA).

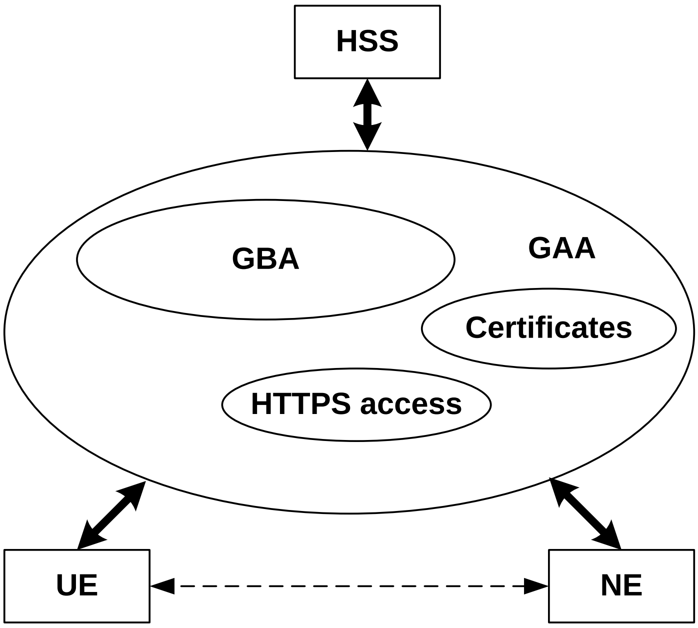


Figure 8 - GAA Architectural Overview

Different 3G Multimedia Services including video conferencing, presence, push to talk etc. potentially use the Generic Bootstrapping Architecture (GBA) to distribute subscriber certificates. These certificates are used by mobile operators to authenticate the subscriber before accessing the multimedia services and applications. Now we discuss components, entities and interfaces of GBA.

Usage of GBA can be divided into two procedures:

- The bootstrapping authentication function
- The bootstrapping usage procedure consisting of authenticating the client to the home network and deriving the key material.

In the usage procedure the User Equipment (UE) tells the Network Application Function (NAF) what key t use. The NAF fetches this key from the Bootstrapping Server Function (BSF). This is depicted in the following diagram.

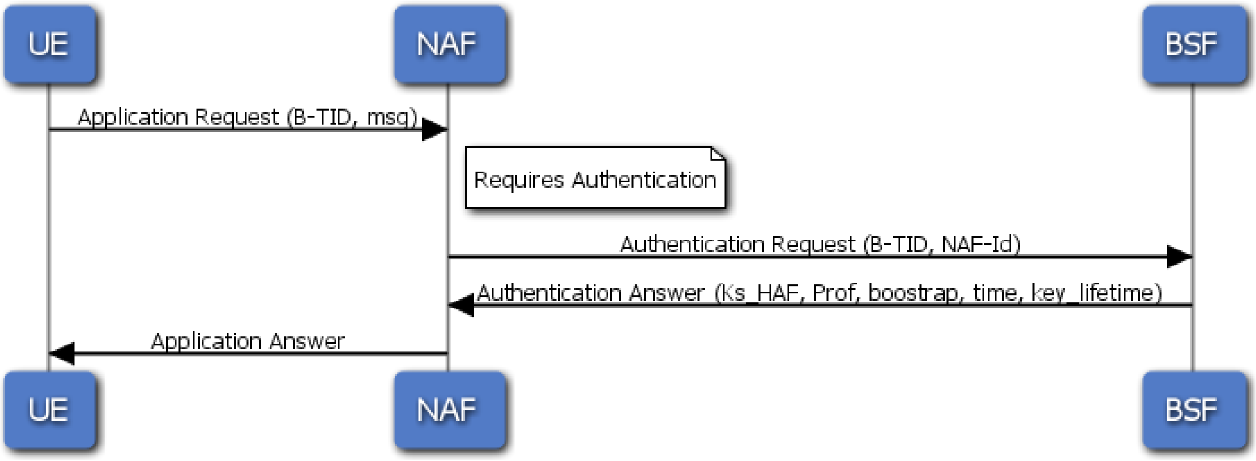

UE
NAF
BSF
Application Request (B-TID, msg)
Requires Authentication
Authentication Request (B-TID, NAF-Id)
Authentication Answer (Ks_HAF, Prof, boostrap, time, key_lifetime)
Application Answer
UE
NAF
BSF

Figure 9 - GBA authentication sequence

There are two different mechanisms for using GBA: GBA_ME and GB_U. GB_U stores the keys on the 2G SIM application. GB_ME is more secure and involves storing the keys to the 3G Universal Subscriber Identification Module (USIM) application of the Universal Integrated Circuit Card (UICC).

6 State of the Art in P2P Identity & Authentication

In this chapter, we describe a number of P2P identity management and authentication mechanisms. For historical reasons many P2P networks have resisted strong identity management schemes as their primary use has been the unauthorised distribution of content. Strong authentication would go against a principle of "plausible deniability" whereby an offending network subscriber could argue that their identity had been impersonated by another party.

However, P2P networks have had to recognise a real problem in content identification and security to protect against malicious incorporation of damaged or virus-infected content. This poisoned content may be pushed into the content overlay by hackers or, in some cases, by commercial entities seeking to reduce unauthorised content sharing on the web. Equally, content communications security has generally been incorporated as a mechanism to obscure the nature of content distributed and to evade firewalls which would drop inspected packets from P2P overlays such as BitTorrent.

6.1 P2P Authentication Schemes

Peer-to-peer (p2p) networks are becoming increasingly popular in recent years due to their distributed and dynamically scalable nature. Although this technology is not yet fully mature, a wide range of possible application are under development by several enterprise vendors in order to take advantage of this extremely useful network tool. So, p2p networks, initially developed for file-sharing, are expected to be used in more sophisticated applications like voice over IP or live video streaming and Video on Demand. Capitalizing this trend, researchers have defined structured and unstructured p2p networks providing a self-organizing substrate for large-scale p2p applications. In most such approaches, the main p2p problems are related to bandwidth management, context management and network scalability. However, due to the extending variety of p2p network applications p2p security has also been elevated into a serious issue.

Making a p2p network secure is a significant challenge. Since the p2p network was not originally designed to withstand an adversary attack it can be easily compromised. Also, due to its distributed nature, this type of networks is subject to additional, more intricate attacks that client-server networks do not face. Without some sort of central management, the p2p network integrity can be endangered by its own network nodes if enough of them decide to behave maliciously. Acting maliciously, a p2p node may lie about his network characteristics, misdirect other nodes that request her assistance or try to take advantage of her role within the p2p network overlay in order to get network resources and provide little or nothing in return (egoistic behavior). Since most p2p networks do not provide any sort of authentication mechanism, the described malicious behaviour attacks can be fairly easily mounted.

More specifically, in p2p networks, apart from traditional network attacks like Man-in-the Middle, spoofing or replay attacks there are a series of attacks that aim in harming the overall network overlay consistency. Such attacks are focused on exploiting the network's Distributed Hash Tables (DHT) and thus disturb the communication between nodes. Among them, the most potentially harmful is the "Sybil" attack [15] where an adversary creates multiple false identities of herself, plants them inside the p2p network and uses them to influence the system behaviour by providing false information to other legitimate nodes. Similar attack is the Eclipse attack [16] where an adversary using her multiple identities conspires to cut off traffic going to and coming from a particular legitimate host thus "eclipsing" her from the network. Both attacks can be used as an amplifier for more intricate attacks like routing or storage attacks [13]. Those attacks target to disrupt the routing DHT mechanism and to provide bogus responses to data queries. Countermeasures against all the above attacks can be focuses on providing an undeniable, fair and unchanged way of binding the identity of a p2p user to a particular p2p node though an authentication mechanism

Authentication is a very intriguing yet complicated subject in p2p networks. As a solution to DHT related attacks, p2p authentication has been addressed by many researchers and a series of technical and research problems have been pointed out [16], [14], [12], [13]. The most widely accepted solution to authenticate peer nodes is to add into the p2p network one or more certificate authorities (CA).

Castro et al in [19] proposes a certificate authority based authentication mechanism that manages to assign nodeIds to principals and to sign *nodeId certificates*, which bind a random nodeId to the public key that speaks for its principal and an IP address. The CAs ensure that nodeIds are chosen randomly from the id space, and prevent nodes from forging nodeIds. Furthermore, these certificates give the overlay a public key infrastructure, suitable for establishing encrypted and authenticated channels between nodes. Certificate authorities can be part of server like structures that can offer a wide variety of services to the p2p network apart from authentication, like content encryption, authorization and accounting.

While CA solution can be effective in centralized or hybrid p2p network, they cannot be easily applied to fully distributed p2p networks since they insert into the system two serious drawbacks. First of all, they can be viewed as a very attractive target since they pose a single point of failure for the p2p overlay. If a certificate authority is compromised then the security of the whole network is bypassed. Additionally, the bandwidth overhead that the certificate authority can add to the system is not neglectible especially in the case of a wide number of peers that require authentication services. In view of a fully functional p2p network where sessions between nodes are constantly established, all authentication requests cannot be handled by CA. Additionally, the scalability of the p2p network is not retained when CA are actively and thoroughly involved in the network communications.

In p2p networks, like live video streaming, as reported in [20] the graph is restructured dynamically in order to be adapted to the underlying network conditions and in peer-node arrivals and departures. By considering this fact and the number of participating peers it is obvious that a solution where every time that two peer nodes become neighbours in the overlay require the communication with a CA that has a centralized architecture hurts the scalability properties of the whole system. For the above reasons, many researchers have proposed p2p authentication schemes that avoid centralized certification and use distributed certificate management [12], [18]. Those approaches focus in finding ways of generating distributed certificates or public keys acting as certificates. The goal is to make each user's public key available to others in such a way that its authenticity is verifiable. To achieve this, the network characteristics of the p2p overlay like reputation, geographic neighbourhood e.t.c., are used. Furthermore, in other approaches identity based cryptography, like pairing, is used for providing distinct identities to the nodes or secret sharing schemes are employed, following threshold cryptography, in order to create a Web of trust through the p2p overlay. Also, zero knowledge schemes are used in order to provide trust, as an alternative to threshold cryptography, that employ during node bootstrapping phase a secret dealer (resembling a CA) [23].

Such a distributed public-key management service based on threshold cryptography is proposed in [22]. The service, as a whole, has a public/private key pair that is used to verify/sign public-key certificates of the network nodes. It is assumed that all nodes in the system know the public key and trust any certificates signed using the corresponding private key. The private key is divided into n shares using an $(n, t+1)$ *threshold cryptography* scheme, and the shares are assigned to n arbitrarily chosen nodes, called servers. Each server generates a partial signature for a certificate to be signed using the private key share that he owns and submits this partial signature to a combiner that computes the full signature from the partial signatures. The full private key can be derived from combining at most $t+1$ private key shares. This mechanism (threshold cryptography) ensures that the system can tolerate a certain number $t < n$ of compromised servers in the sense that at least $t + 1$ partial signatures are needed to compute a correct signature.

Those solutions do not always lead to successful results since they may introduce additional bandwidth overhead or not fully protect against Sybil attacks (a sufficiently big number of node multiple identities can still compromise the p2p overlay) [15], [17].

7 P2P-IMS Authentication Challenges

P2P-Authentication shall enable peers to exchange authentic messages directly between them. In an open P2P System (e.g. BitTorrent), users are anonymous so that no proper charging or service offering is possible due to this fact. Even worse, many possible malicious scenarios are thinkable to the harm of the user. In this chapter we will list and describe the most important attack scenarios in open P2P networks.

7.1 Sybil Attack

A malicious Node can join an overlay and impersonate several different identities, i.e. he is present in the overlay with multiple IDs, also in terms of network-address and port number. Other peers may think that they are talking to different peers, but in fact talk to only one single node. This way, an attacker may gain control over a large part of the overlay. With an increasing number of identities, the probability for an honest node to contact the malicious node rises significantly. The malicious node's chances for further misbehavior (e.g. on routing) also increase, based on the overlays distribution algorithm and related routing functions. For a reputation based system, which determines the trustworthiness of a node by some kind of vote, such an attack can result in a catastrophe, affirming the malicious node's false identity.

7.2 Bootstrapping

When a node enters an overlay, it does so by querying an already known node, which is in the overlay (boot node). This node will provide the new node with information on how to join the overlay (e.g. which other nodes are its new neighbours, etc.) The bootstrapping attack now simply assumes that the boot node is malicious and gives the new node only neighbours, which are already under its (the malicious node's) control. This makes the new node the only node in a private overlay, filled solely with malicious instances of the malicious node. The results are the same as for a Sybil attack with a huge number in malicious identities.

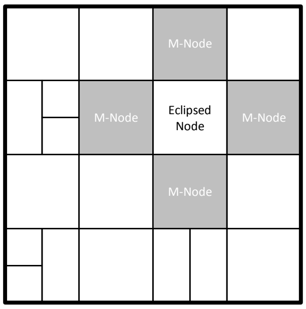
7.3 Information falsification

Information falsification is a problem in a P2P system, where Information or content is stored on foreign nodes (e.g. in a DHT). A writer node stores the information in a storage node, which might be malicious. Without security measures, a reader node which requests that information from the storage node, might receive information which is different from the originally stored one, as the storage node can modify the information arbitrarily. Thus, sensitive information, which must not be corrupted, cannot be stored in a distributed way without additional security mechanisms.

7.4 Eclipse attack

The purpose of an eclipse attack is to isolate a victim node from the overlay so that no more message routing from or to that node appears. In order to achieve this, the attacker impersonates all possible direct neighbours of the victim in the overlay (comparable to a directed sybil attack, s.a.)

The eclipse attack can only appear, if a malicious node has an influence on its own position in the overlay, i.e. if its neighbor nodes cannot verify and reject its request to join at that specific point in the overlay.



M-Node = Malicious Node

Figure 10 - Eclipse Attack in a 2D Content Addressable Network

The eclipse attack is different from the bootstrapping attack, as it can appear while the victim node is already part of a healthy P2P overlay, while the bootstrapping attack must occur during the joining phase of the victim node.

7.5 Conclusions

One of the prime challenges of VITAL++ is to enable secure communication between peers in a way that attacks like the depicted ones cannot occur. As the depicted attack scenarios have shown, the most important problem is that peers do not know to which other true identity they are talking to when exchanging messages. All the above attacks would not be possible, if overlays were planned by a trusted entity and if peers could determine the true identity of the peers they communicate with.

8 State of the art in Accounting

8.1 Service Environment and Accounting Standards

Over the past number of years the world of service provisioning has changed rapidly with the advent of new technologies. The Next Generation of service infrastructure is opening previously unseen avenues in service provision and user subscriptions [8]. The service world is going (has mostly gone) mobile. Smart phones have replaced the PDA and have swept the consumer market to a state of near omnipresence. Developers are now capitalizing on these new platforms and of course the available bandwidth of 3G and WiFi networks that enable a far richer variety of services to be created and delivered to mobile users. Handsets themselves are being deployed with SIP capabilities and are increasingly more directed towards acknowledging the huge peer-to-peer market of legal file distribution, messaging, and VoIP. Providing services for these mobile devices is now a strong and competitive market place that is set to continue growth over the coming years.

Traditionally, new services have always been slow to emerge onto the market. This is largely due to operator restrictions that curb the flow of service delivery through rigid quality process and restrictive barriers to entry such as financial burdens on the service developer. The internet and associated technologies has changed significantly in recent years with the advent of IP Multimedia Subsystem (IMS) and the potential for an explosion in IP-based service providers now exists because of more programmable networks and increased facility available for more formal and richer peer-to-peer service provisioning.

> *"Unlike access, premium services cannot be profitably priced at a flat rate; they must be priced by usage... without a robust, flexible rating infrastructure, IP service providers cannot capitalize on IP protocol advancements to offer and bill for the premium services"*
> – A. Heintz and Dr. M. Lucas [4]

The above quote was taken from an article “IP Pricing and Rating”, featured in Billing World Magazine in June 1999. Despite being printed more than 10 years ago, the essence of the quotation is still very much applicable to the current environment of services. We have become accustomed to flat rate internet access through WiFi access points or 3G price plans but with exception to the free services that account for the majority of services running over these carrier networks, there still exists the problem of how to account for and dynamically adjust to usage of premium services. Flat rate charging does not work for end-users on a per service basis and as such it does not work for service providers either because it reduces uptake of their services.

The main standard governing any accounting system development and operation has its roots in IETF's AAA standard [3]. The IETF's AAA (Authentication Authorization and Accounting) Working Group is focused on the development of functional requirements for usage mediation, revenue generation, billing, service quality and reliability, and roaming. AAA defines the system in IP-based networking to control what computer resources users have access to and to keep track of the activity of users over a network. The AAA specification has, in recent years, moved towards a fuller requirements specification known as A4C (Authorization, Authentication, Accounting, Auditing, and Charging). The addition of auditing for security and fraud detection is wholly necessary in public service provisioning however, the majority of rating/accounting system design can be adequately covered by the AAA specification guidelines.

8.2 Rating Systems – Past, Present, and Current

Typically there exist 3 main types of rating systems used by service providers to account for service usage. These equate to Program-code-based systems, Rule-based systems, and Table-driven systems.

8.2.1 Program-code-based Rating Systems

Program code-based rating is the method used for the earliest rating software that was implemented. There are companies still running this type of software from more than twenty years ago. This kind of rating is normally the fastest in production because it involves little disk I/O and is compiled coderather than interpreted code. Program code-based rating must be created and maintained by programmers. Its problem is a side effect of the behavior of programmers. Most programs begin with a coherent style and structure. Provided the same programmer works on the software, the style and structure remains consistent and the costs of changing it remain low. The reality of software development is that new programmers often take over development and maintenance from the original programmers.

When your rating process is encoded in a formal programming language, it is difficult for each new programmer to understand the meaning of the algorithm. The difficulty encountered could be due to a number of reasons such as unfamiliarity with the programming language, a particular programming style used, or very badly written code. Code complexity is a serious concern for software developers and very complex programs are considered to be of considerable risk to the maintenance of a system [7]. This difficulty increases the amount of hours that you must spend before you can confidently change your rating process. When a change is requested, each programmer must first understand the code and then change it. Most programmers are under pressure to do their work quickly. The amount of work involved in confidently understanding the existing rating program code causes programmers to insert code that is not coherent with the existing code. The programmer puts in code that is certain to get the desired results but inadvertently increases the complexity of the code. Each time that additional non-coherent code is added, the program complexity multiplies.

8.2.2 Rule-based Rating Systems

Rule-based rating is program code-based rating reborn. In this case you have a structure, the rating engine, where a rating-orientated programming language is hosted. This is a simple language focused on the business problem. Usually a business/tariff analyst can learn the language in a relatively short period of time compared to the training needed to develop skills in a conventional programming language. Each rating plan can be programmed in a higher-level language orientated toward the rating process. This means that rating plans can take advantage of well-tested underlying code (like a table-driven rating process) while defining the rate plan in a language that is easier to maintain and also helps to decouple the rating plan from the core system functionality. Rule-based systems maintain a rules database of business logic that is applied to usage records as they are received. Unlike program code-based rating, the rules are often contained in a non-executable format and require the interpreting functionality of another component in order to be applied to the usage data.

Rule-based rating uses more computer power than code-based rating. Program code-based rating is compiled whereas rule-based rating is interpreted. This performance difference has become negligible, despite ever increasing transaction loads, because the power of today's computers has increased so dramatically from earlier machines. This makes it possible for the rate plan development to be done by business/rating analysts rather than programmers. By extending the set of people involved in constructing and maintaining the algorithms from just programmers (who seek skill development and opportunities to develop cleverly sophisticated programs) to include business or rating analysts (who value order and control and who are more interested in business goals), the problem of the gilded edge becomes less of an issue for those who wish to experiment with the algorithms. The amount of training needed to manage the rate plans and rating processes is also reduced. This investment in technology (rule-based rating), affords the ability to solve many organizational problems with respect to freedom to experiment with algorithms that can operate in a sandbox and without the necessity of programming experience to modify them. Successful rule-based rating solutions include FusionWorks ActiveRate from Openet Telecom [9].

8.2.3 Table-based Rating Systems

Table-based rating is the process of rating transactions where there are one or more tables of data that encode different rating cases and the amounts to use in calculating the charge. These tables are stored outside of the program code so that the rating behavior can be changed without the program being modified.

Over the last twenty years, many companies have sought to make it easier to control the difficulty presented by program code-based systems (i.e. the lack of separation of tariff schemes from the program logic) by encoding their rating algorithms in tables. This permits you to encode the parameters of how to rate transactions in the table. Process code still exists but its functional meaning is more clearly defined and it is easier to program and maintain. If you need to add a new rate plan, most of the time you can encode it in the table. This means you still change the program code but less frequently. Fewer changes mean that the code remains coherent for longer and lengthens the lifetime of the rating code. You also are more likely to be able to define the functionality of the new program code clearly. As such table-driven rating was seen as a big step forward for the industry.

Each column in the table represents a data element that is needed for the rating process. Each row represents a rating case that can apply. The problem that arises with table-driven rating algorithms is the static size and form of the tables. They pose a major problem when dealing with the continuing changes in rating plans. As marketing organizations struggle to find new algorithms that connect with customer value, they make up new methods of rating and then discard them when they fail to capture the market's attention. With each new way of rating, new kinds of information are needed in the table and new columns are created in the table. Since all of the rating code uses the table, you end up with changes needed in all the code when you add new columns.

Furthermore, columns are often added to tables to accommodate current needs but then become redundant at later stage. These columns are not removed for fear of dislocating other rate plans or program code to assure backwards compatibility. This needlessly increases the size of large tables and complicates the maintenance of data unnecessarily.

8.3 Usage Data Representation

It is important to avail of structured usage data that is logically interpretable and also atomically meaningful. Obviously no single usage parameter on its own will ever present this but usage records, a collection of usage data, can appear in many different formats. Traditional CDR (Charge Detail Record) type records are presented in many different formats including comma separated value (csv) files and ASN.1 [6] encoded byte fields. Neither of these formats is particularly meaningful for anyone reading the files nor do they have any obvious structure to an individual inspecting the file. A guide that identifies the sequence of information and what element of each CSV line contains is required in order to interpret the data. This section will detail the advances made, in usage data representation, by an organization known as the IPDR.org whom have the backing of major industrial corporations and leading experts in the domain.

The IPDR (Internet Protocol Detail Record) organization is an industrial consortium, founded by some of the prominent vendors providing management solutions for IP-based networks. Members include Hewlett-Packard, Oracle, Portal, Sun, AT&T, Amdocs, Compaq, XACCT, Aptis, Andersen Consulting, CableData, Clarent, Narus, Savera, and TeleStrategies. The primary objective of the IPDR organization is to specify the essential attributes of information exchange between network elements and services, OSSs and BSSs. This resulting specification provides the foundation for the development of open, carrier-grade support systems that enable next- generation networks and services to operate efficiently and cost effectively. The IPDR organization has adopted the core functional roles and interfaces of the TM Forum's TOM [1] for the specification of interfaces between OSSs and BSSs. The specific goals of the IPDR organization:

- Define an open, flexible record format (the IPDR structure) for exchanging usage information
- Define essential parameters that can be used to define a service or network usage
- Provide an extension mechanism so network and service elements, and support systems can exchange optional usage metrics for a particular service
- Provide a repository for defined IPDRs

The idea central to the IPDR initiative is similar to that of the Charge Detail Record (CDR), which is a record of system events and is widely used in the telephony world. A CDR is produced every time a user makes a call. Among other information, a CDR contains the start and end times of calls, and the identification of the calling and called parties. This information is then used to create accounting records that support bill preparation and subsequent analysis. The IPDR is the corresponding record for IP-based networks.

The IPDR organization has produced the Network Data Management-Usage (NDM-U) specification [5] for the detail record that tracks network and service usage and facilitates value-based billing for IP-based services.

When deciding upon a format for the document, the IPDR organization chose XML as a desirable mechanism of marking up usage data. The IPDR format capitalizes on the benefits offered by XML for data interchange and ease of interpretation. The XML record structure and service definitions provide a means to begin representing service usage information in a consistent, self-describing, human readable format. These structures called IPDRs allow for the creation of documents by one system in a format that can be understood and easily used by another. The IPDR is capable of characterizing any type of service usage that could be collected from an IPDR compliant network and service. In order to achieve this, the IPDR has been broadly designed around five attributes common to all records. These components are the, who, what, where, when and why values that describe a particular usage event. Furthermore, the richness of the IPDR allows for easy capture, from the service level, of data relating to individual users within a single IP address multi-user environment. Previously, accounting for services was conducted purely at the network level and distinction between users was carried out on IP address alone [10]. However, it can be seen that this is no longer a feasible approach to modern internet-based services and the IPDR facilitates the capture of this important data at the service level.

Each IPDR is encapsulated in an IPDR Document (IPDRDoc), which is the unit of information exchange, and contains one or multiple IPDRs. A single IPDR Master Schema Document declares elements common to all IP-Based services. The NDM-U proposes to define an IP-Based Service Specific Schemas for existing and emerging services. The IPDR organization has defined service specific schemas for services including Email, Video on Demand (VoD), Voice over IP (VoIP) and Internet Access. The IPDR Document hierarchy allows an IPDRDoc to contain many usage records (IPDRs). The IPDRDoc structure is presented in Figure 11.

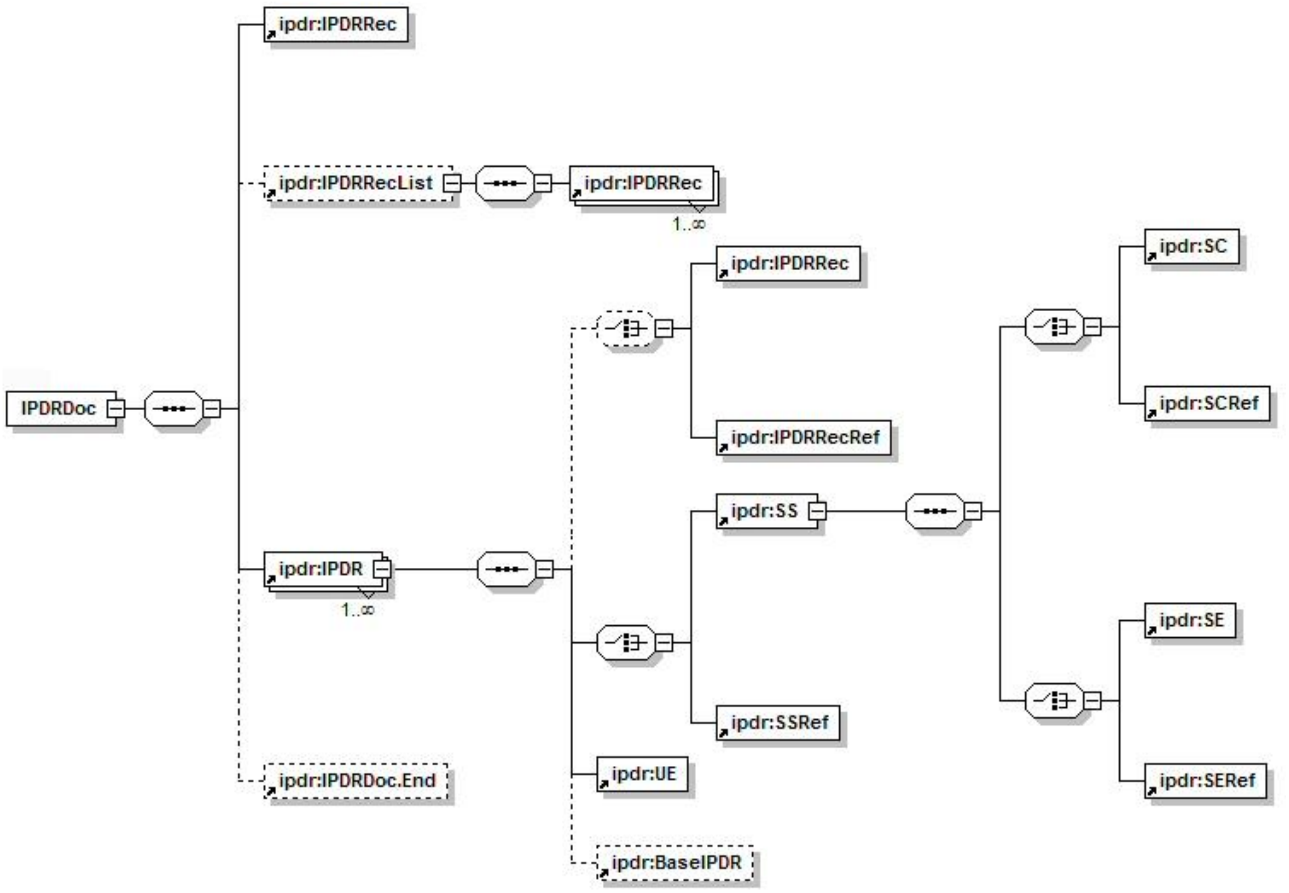


Figure 11 - IPDR Structure

The structure of the IPDR document is broken down into an element hierarchy rooted by the IPDRDoc element. The IPDRDoc can contain a number of IPDR elements (each representing a service session's usage data). The IPDRDoc can also, optionally, contain information about the IPDRRec (IPDR recorders) used to gather the data and also an IPDRDoc.End element to act as a signaling element as the end of the IPDRs being sent. Within the IPDR element itself the structure is reasonably flat. A SS (service session) element describes the service with respect to its type and user information. The main stay of the data is contained within the UE element (usage element) that contains unlimited, service provider-defined elements to detail information such as session duration, data transfer size or whatever set of data best represents the particular services being recorded.

8.4 Conclusions

The state of the art in accounting/rating revolves around established and robust implementations of typically, operator grade systems with a glimmer of reform and modernisation shown through the emergence of the IPDR standard for usage data mark-up.

Rating systems are still telco-orientated and as such provide an inhibiting cost factor for smaller, would be service providers. These smaller service providers have a potential market but require lesser overheads and more dynamic, lightweight accounting solutions to match their business needs.

The establishment of a carrier network and mobile handsets with advanced peer-to-peer protocol support and enriched user interfaces means that a great number of services are now awaiting deployment. Most mobile applications have shown that the market is driven by premium applications and not premium services, i.e. the cost is an upfront charge for the application downloaded by the user and not an ongoing monthly or per use charge associated with the underlying service. Thus a mechanism of rapidly and dynamically adapting to the accounting needs of these service types is warranted.

Vital++ will implement an accounting solution that matches the requirements of one such service scenario matching the qualities outlined above, utilising the IPDR standard for usage data exchange, and coupling a custom developed rating engine that adheres to a lightweight and easily adaptable design model.

9 Overview of VITAL++ architecture

The VITAL++ overall architecture is basically distributed over the Client and the NGN/IMS area of functionalities. In this chapter we will give an overview over the whole architecture, which will then be discussed in more detail in the subsequent sections.

In order to address the VITAL++ challenges, multiple sub-architectures have been defined, which interact among each other. These are the P2P Authentication sub-architecture (P2PA), the Content Index sub-architecture (CI), the Overlay Management sub-architecture (OM) and the Content Security sub-architecture (CS). Each sub-architecture spans over the client, the network and the IMS with its components. Sub-architectures may interact among each other in an arbitrary way, especially in the client, while on the NGN side there need to be well defined interfaces. Thus the media exchange is not entitled as sub-architecture, but it interacts with these and itself in the same as well as in remote clients.

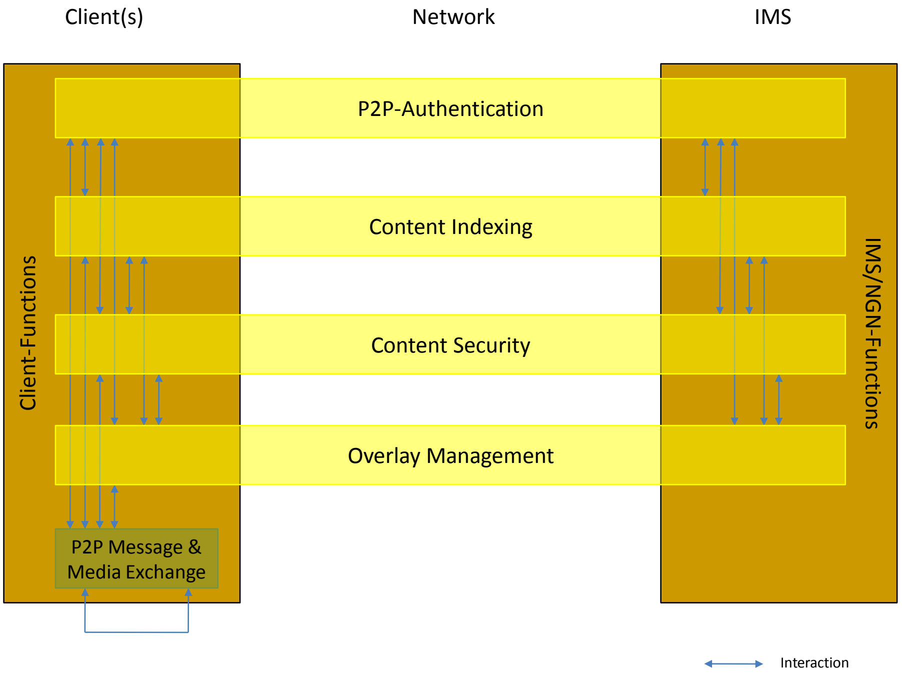


Figure 12 - VITAL++ abstract view of the overall architecture

9.1 Client

The terms "Client" and "Peer" are used equivalently in this document as they refer to the same thing. The VITAL++ client is a hybrid client. This means it is an IMS client and a P2P client at the same time. The IMS functionalities are used to mainly interact with an IMS core or system for exchanging control information, while the P2P part is used to exchange content with other peers.

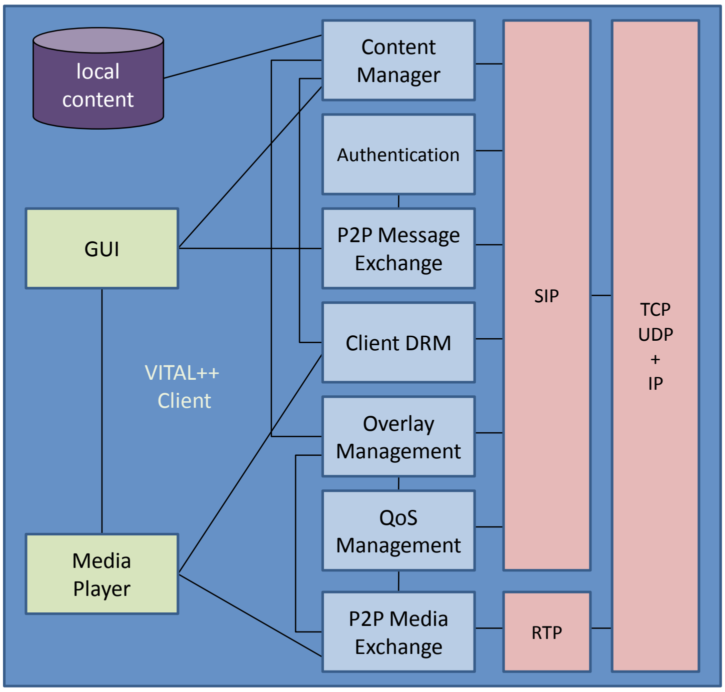


Figure 13 - Client functional blocks

The components of the client are directly derived from the necessity to interact with other clients and the IMS core in order to fulfil the envisaged features. Figure 13 illustrates the functional blocks inside the client. These are the content manager, which is responsible for publishing and discovering content as well as triggering DRM operations via the client DRM module if a licence needs to be obtained. The authentication module obtains and manages certificates of VITAL++ entities (clients, application servers, root-certificate). It interacts mainly with the P2P message exchange in order to sign and verify messages. The latter has the purpose to exchange P2P messages with other peers for genric purposes (i.e. playlist exchange, etc.). The overlay management module obtains overlay changes from the application server and re-organizes its neighbourhood accordingly, also to respect to QoS requirements, issued by the QoS management module, which can also realize QoS enforcement via NGN mechanisms. Also standard IMS client functionality is realized (not depicted) for initial IMS registration and IMS session management.

9.2 Platform

The platform side of the architecture consists of four application server entities, which can be co-located in the same box (as depicted), or distributed over several machines. The communication with the client occurs mainly through the IMS core and its call/session control functions (P/I/S-CSCF). Each of the functional blocks in the application server refers to a related sub-architecture.

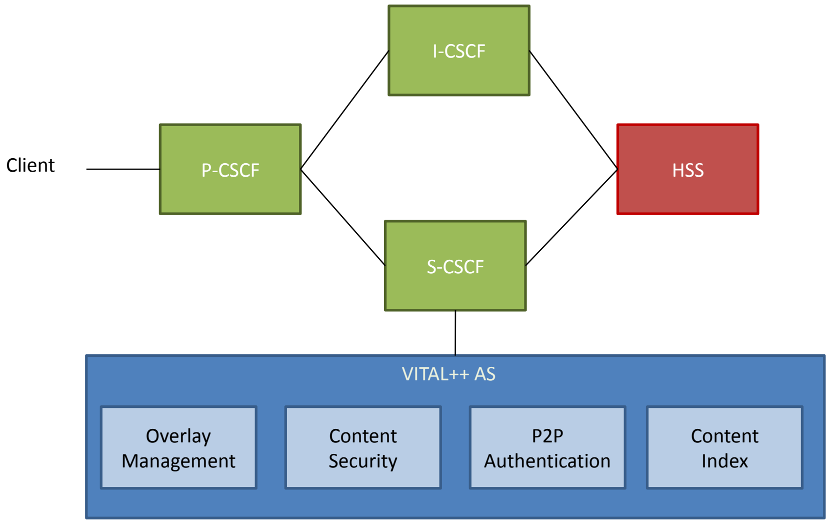


Figure 14 - Platform components

Figure 14 depicts the platform components and their relation with other IMS objects. The functional blocks are the

- P2P-Authentication module, which stores client certificates for use by other modules, serves the client with initial credentials and signs the client's certificates on request.
- Content Index module, which stores content descriptions and metadata and provides search functions to the clients.
- Overlay Management module, which constructs and maintains optimised overlays according to the client's connectivity.
- Content Security module, which provides and maintains DRM licenses for published content.

10 Analysis of P2P authentication

In order to address the introduced challenges in P2P authenticity from chapter 7, the P2P-Authentication sub-architecture (P2PA-SA) has been defined. The P2PA-SA works with certificates, i.e. digitally signed chunks of data, which describe an entity and its properties, e.g. its public identity, public key and access rights. In the VITAL++ scope, three levels of certificates are distinguished, as shown in the following table.

Root Certificate	Self-signed. Pre-installed in every client and P2P-Authentication server module (CA).
Server Certificate (CA Certificate)	Signed by Root-CA. Pre-installed in every P2P-Authentication server module. Describes the identity of the server domain and its public key. Acquired by each client during registration.
Client Certificate	Personal certificate, created by each client Signed by a P2P-Authentication server module on request (CA). Describes at least the public identity of the client and its public key.

Table 1: VITAL++ Certificate Types

Finally, each client is equipped with these three certificates, which allow it to perform all authenticity transactions and checks as required by client homed applications.

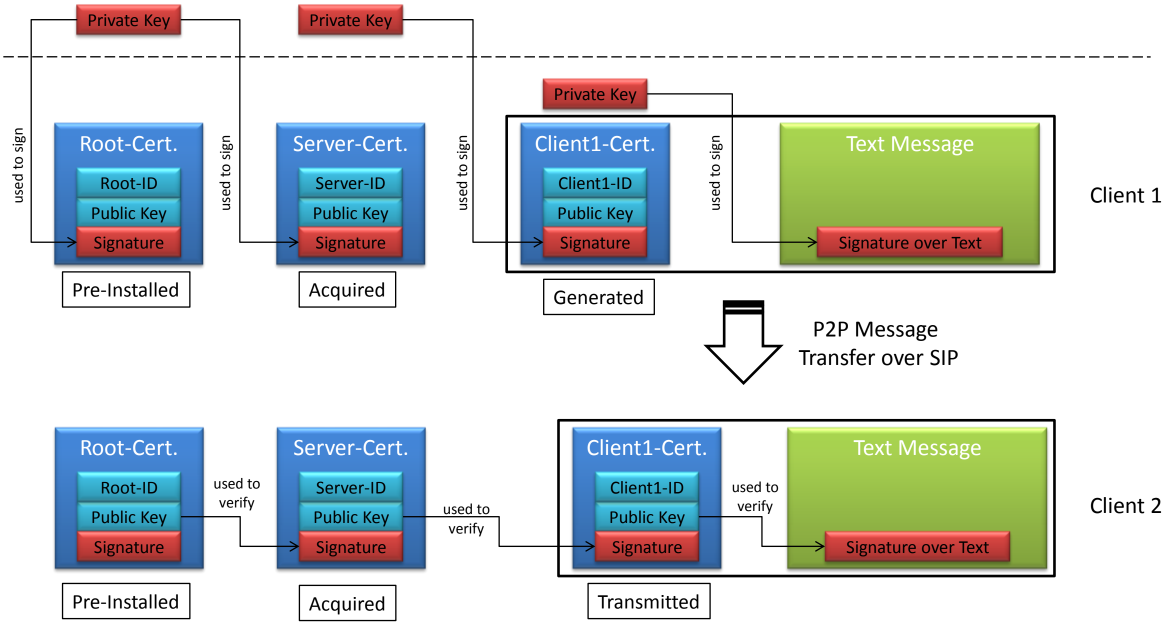


Figure 15 - Relation between Certificates and Messages

The relation between the certificates and their use in order to enable authentic message exchange is depicted in Figure 15.

The peer key pair authentication and certification general idea is as follows, assuming that peer A has to authenticate its public key with peer B using a Certificate authority CA. Note that CA is has a public – private key pair itself.

However, in a p2p distributed network, certificate authorities should be used cautiously. They can be a bottleneck for the functionality of the whole network from bandwidth resources and security point of view. Thus, their use should be minimized to retain the p2p network scalability and efficiency by introducing a scalable authentication mechanism

10.1 Scalable p2p authentication Scheme

The increased applicability of peer to peer networks, introduces new challenges in such networks' functionality and principles. Peer to peer security constitutes such a challenge that stems from the need to constrain or eliminate malicious peer behavior within the network overlay and to authenticate node users without losing the scalability of the network or overusing the network resources (e.x. bandwidth). In this work, an authentication scheme is proposed based on Certificate authority public key mechanism that manages to retain network scalability and addresses known peer to peer network security problems like Sybil attacks. The proposed scheme is applied on DHTs peer to peer networks and comprises of two functions, new node authentication and node to node authentication. In the first function a new node becomes part of the peer to peer overlay by obtaining a unique identity and a zone in the network's DHT that is certified by a certificate authority through certificate issuing. We propose a methodology for retaining the secure storage of the certificate in an authenticator node that the DHT neighbors of the new node vouch for. This vouching is done by issuing a securityID number by introducing a security DHT. Certificate storage is performed by using a node's secuirtyID to find the zone of the authenticator node. Node to node authentication is performed when a session between two nodes is initiated. An Authentication session protocol is proposed for this reason that uses the authenticator nodes as certification advocates between the session two involved nodes. Security analysis of the system reveals that the proposed authentication scheme is resistant to generic and peer to peer specific attacks and performance results reveal that it does not impose any important bandwidth overhead to the network.

10.1.1 Distributed Hash Tables (DHT) - Basic Concept

DHTs are distributed systems and their functionality is to perform the distributed maintenance and use of a database. In general every DHT forms a virtual space. Every node that enters the system is responsible for a portion of this virtual space. Through randomized insertion in this space the nodes that form the database are able to balance the size of the zones of the virtual space among them. In Figure 16 we see a DHT formed by four nodes. The two dimensional virtual space is divided in four equal parts and each node holds the one fourth of this space.

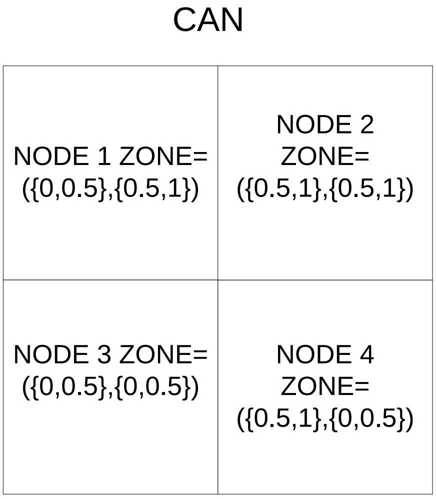

CAN
NODE 1 ZONE= ({0,0.5},{0.5,1})
NODE 2 ZONE= ({0.5,1},{0.5,1})
NODE 3 ZONE= ({0,0.5},{0,0.5})
NODE 4 ZONE= ({0.5,1},{0,0.5})

Figure 16 - An example of A DHT called CAN formed by four nodes.

Each node also has a list of network addresses that is called routing table. Routing tables providing a distributed data structure that a routing algorithm uses and through this it is able to it is able to find the network address of the node that is responsible for a point in this virtual space. In CAN every node hold the network addresses and the zones of the nodes with zones adjacent to its. For example in Figure 16 the routing table of node 1 consists of the network addresses and the zones of node 2 and node 3.

The properties of every DHT that also our system inherits are: the balance of network load that routing process introduces among the participating peers, the automation of the database and routing table reconstruction to node arrivals and departures, The low latency routing process that has a logarithmic relationship with the number of nodes that participate. In this way we have a self- organized and scalable system with high performance and so it is suitable for our system

10.1.2 Authentication Model

Each peer needs to have a solid way in order to be authenticated and integrated to the overall system. This integration might involve communication to other peers and communication to server entities. In all forms of those communications, the identity of each peer should be authorized so that trust among them can be established. While password authentication might seem an adequate solution, the authentication that this solution provides is considered very weak. Strong authentication is achieved through the use of public key cryptography (digital signatures) and hash functions.

Each peer during authentication must be able to reliably verify the identity of another peer. This can happen by asking the other peer to provide some credentials proving the authenticity of its identity. Only then can the two peers trust each other and exchange data, like video blocks. However, how can a peer trust that it communicates with peers that are what they claim to be? This authenticity can be provided by a third party trusted authority known as certificate authority. The role of the certificate authority (CA) is to issue specific certificates for the identity and characteristics of each peer. The phases involved in such an authentication scheme are three:

Registration: At this point a new peer wants to enter the p2p network and needs to communicate with the CA in order to acquire a unique identification number (nodeID) and certify its public-private key pair. The certificate authority is responsible for providing the new node with a nodeID number and certifying the node’s generated Public-private key pair. At the end of the registration phase, a new peer is uniquely and undeniably associated with a public – private key pair. These information are included in the issued certificate in an undeniable and unchanged way.

Authentication: The Authentication phase is required when a node (node A) information is added to the DHT. In that case, a request is formed from the node (node B) occupying the DHT zone in which A wants to enter. This request is about authenticating the identity of the node that is inserted to the DHT. This insertion is granted only when node B, that is already involved in the DHT (and therefore is authenticated), verifies node's A credentials (the certificate and nodeID) are verified.

Linking: We perform linking only after a successful authentication phase. In that case, we insert node A into the DHT and it becomes also an authentication entity. Only after a successful linking, is a new peer fully authenticated and can be considered part of the p2p network overlay.

10.1.3 Registration Phase

The registration phase is initialized by a request from a new node to be associated with the p2p network overlay in order to offer and receive this network's services (live video streaming). The p2p network candidate new node needs to generate a public - private key pair and submit the public part of the key as a certificate request along with its IP address. However, the new node needs to sign this public key with its private key. This operation is required for data integrity and for letting the CA know that the new node does indeed know the private key.

Thus, the CA verifies the signature using the new node's public key and then generates an identification number (nodeID) using the IP address of the node. The resulting number is used for issuing a certificate for the node's public key. The certificate has an expiration date and includes the nodeID value and IP address of the new node. The CA is also responsible for generating the DHT zone where the new node will be added. These coordinates are generated using an appropriate HMAC function where the nodeID and public key is used as input and a secret value known only to the CA is used as key. The resulting DHT coordinates (zone) are added to the certificate and the whole certificate is encrypted by the CA using its private key.

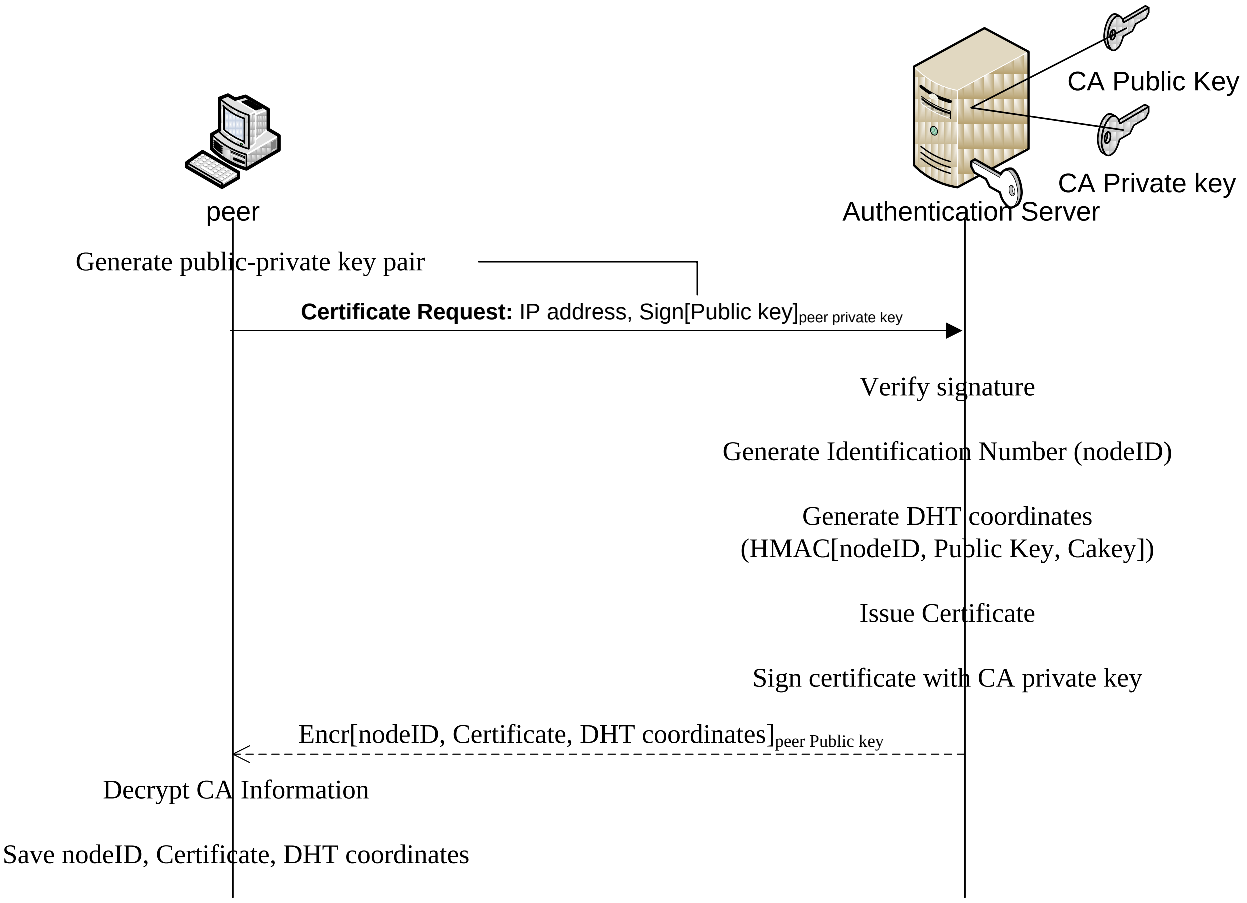

CA Public Key
CA Private key
peer
Authentication Server
Generate public-private key pair
Certificate Request: IP address, Sign[Public key]$_{\text{peer private key}}$
Verify signature
Generate Identification Number (nodeID)
Generate DHT coordinates
(HMAC[nodeID, Public Key, Cakey])
Issue Certificate
Sign certificate with CA private key
Encr[nodeID, Certificate, DHT coordinates]$_{\text{peer Public key}}$
Decrypt CA Information
Save nodeID, Certificate, DHT coordinates

Figure 17 - The proposed registration phase message exchange.

The final step to the registration phase is to send the authentication information to the new node. These information consist the nodeID, the encrypted certificate and the DHT zone of the node in the *d*-dimensional virtual space of the network overlay. Those values are concatenated, are encrypted using the new node's public key and are transmitted to the new node. The final encryption ensures that the certificate can only be decrypted by an entity that knows the associated private key. This renders a Man in the Middle attack useless. Also, the encrypted certificate can be easily decrypted by the new node using CA public key (it is considered known) but it cannot be altered. The Registration phase protocol is presented in Figure 17.

This certificate can be used along with the node's public key during authentication with other nodes so as to verify that the transmitted public key is not forged or altered by an eavesdropper performing spoofing or replay attack. The information provided by the CA are necessary in order to make the new node part of the network overlay enabling him to acquire a DHT zone. Note, that the certificate can also include authorization and accounting information.

10.1.4 Authentication Phase

Authentication, using the proposed in this paper methodology, is required when a new node that has acquired all certificate information during the registration phase enters the system.

In our p2p system, the overlay is based on DHTs for creating a virtual *d* dimensional coordinate space. Each node is associated to a specific zone marked by specific *d*-dimensional coordinates that are initially provided by the CA. When a new node (node A) requests a zone within the DHT overlay, it needs to authenticate itself to the node (node B) that already occupies this zone. Only after a successful authentication can the zone split and the new node acquire routing tables and security information through the DHT. The authentication sequence does not involves communication with the CA since node B can confirm that node A has legitimate credentials provided by the CA by knowing the CA public key. The Authentication protocol involves one way authentication (only node A needs to be authenticated) since the presence of node B within the DHT verifies that this node has already passed a similar authentication sequence. The Authentication protocol has the following form, also described in Figure 18.

1. Node A generates r_A, which is a non-repeating number that is used to detect replay attacks
2. Node A send node B the following: *{r_A, t_A, nodeID$_A$, nodeID$_B$, certificate$_A$, A-pub.key, signed Data}* where t_A is a timestamp indicating the expiration time of the transmitted message, *nodeID$_A$*, *nodeID$_B$* is the identification number of A and B accordingly, *A-pub.key* is the public key of A and *certificate$_A$* is the certificate of A. Additionally, *signed data* is a digital signature, using node *A private key,* of the whole transmitted message (data) needed for data integrity.
3. node B checks that node*ID$_B$* belongs to himself and therefore is the intended information receiver
4. node B verifies the certificate of A using the CA public key
5. node B verifies the *sign Data* and thus the integrity of the sent information and that the they were truly sent by node A.
6. B checks if the timestamp is up-to-date
7. B checks if r_A is replayed

Note, that in order to verify the certificate of A, node B needs to have some mean of reading the data in the certificate (the certificate is always encrypted using the CA private key). We can make a safe assumption that the CA characteristics are well known to all authorized nodes of the p2p network overlay. Those characteristics involve the CA public key and nodeID. Thus, verifying the certificate's authenticity becomes a fairly easy task. Verifying node A transmitted certificate involves using the CA public key to decrypt the certificate and read it. The data in it must confirm the transmitted by node A values.

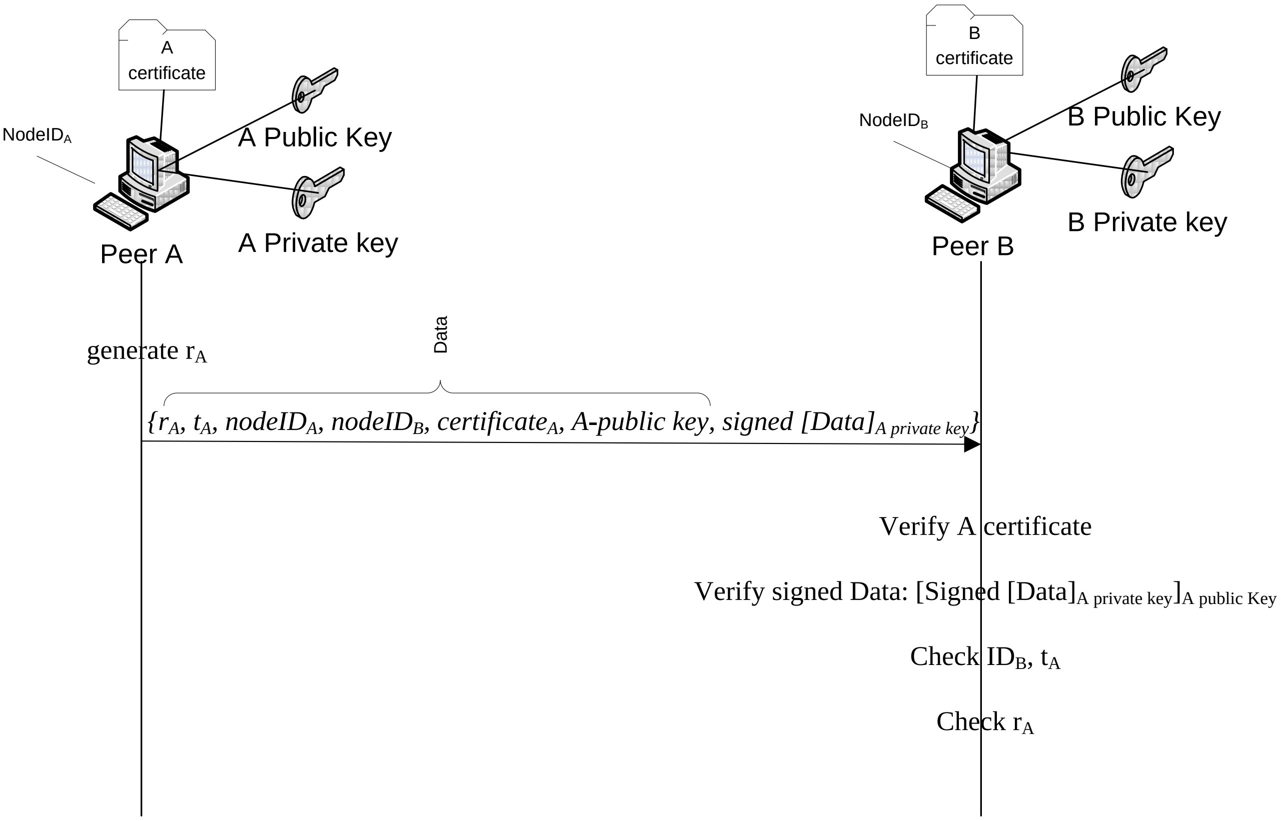

A certificate
A Public Key
NodeID$_A$
A Private key
Peer A
B certificate
NodeID$_B$
B Public Key
B Private key
Peer B
generate r_A
Data
{r_A, t_A, $nodeID_A$, $nodeID_B$, $certificate_A$, A-public key, signed $[Data]_{A\ private\ key}$}
Verify A certificate
Verify signed Data: $[Signed\ [Data]_{A\ private\ key}]_{A\ public\ Key}$
Check ID_B, t_A
Check r_A

Figure 18 - The proposed Authentication phase message exchange.

10.1.5 Linking Phase

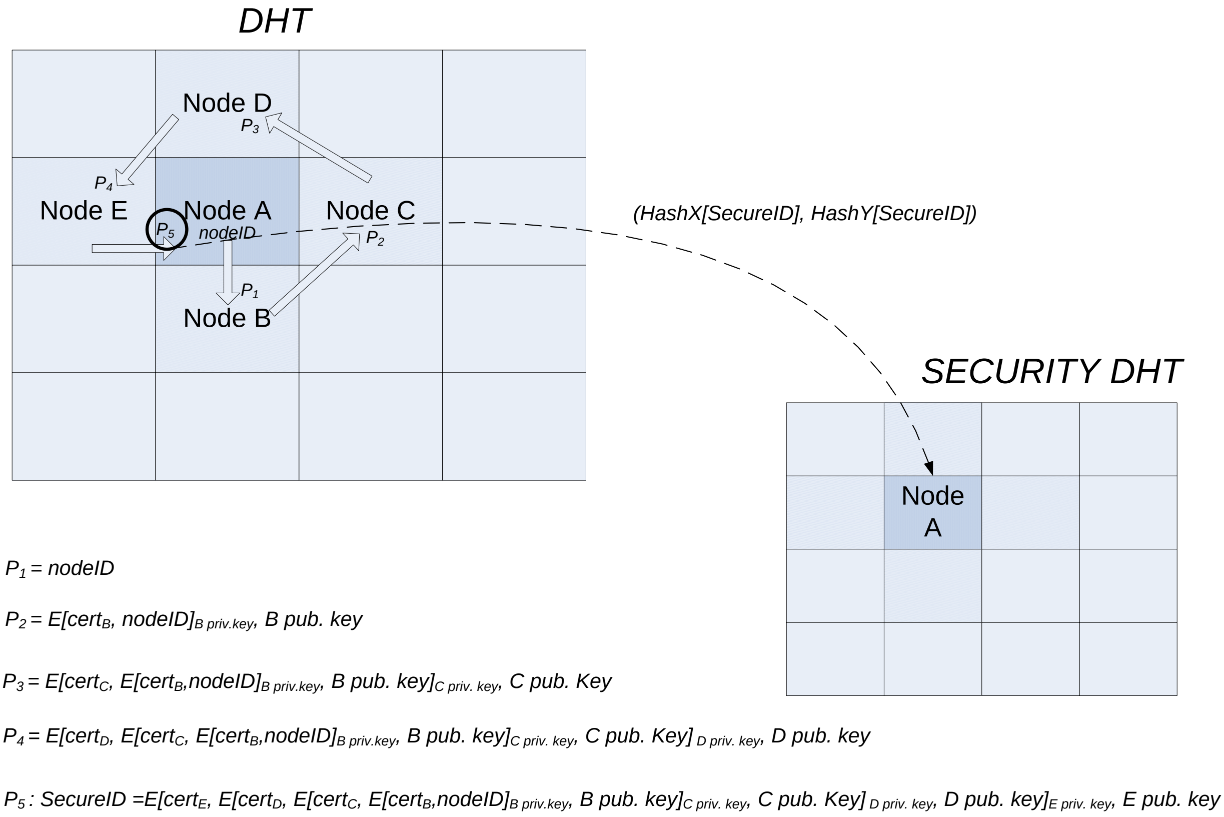
After authenticating a new node (node A) with the node (node B) occupying the DHT zone assigned by the CA, the zone itself is split in order to make the required space for node A within the DHT. Nodes that have zones adjacent with the zone of node A (including node B) are addressed as neighbor nodes and have to be notified about the existence of a new authenticated node in their neighborhood. However, these operations are only part of the full authentication of node A. The successful authentication of the new node must be propagated to the node's neighbors and future authentication of this node needs to be ensured in an optimized way (i.e. minimizing the interference of the CA).

After the authentication phase, a node has no practical control of its certificate. Its certificate is copied to another node of the p2p overlay (node C) that can act as a certificate verifier (authenticator) when asked. To achieve this, we propose the use of a security DHT that is responsible for handling the certificates of each authenticated node within the p2p overlay. The zone's virtual *d*-dimensional coordinates of each node within the security DHT can be found by hashing a special value, called *securityID*. This value is found by a collaboration of all node's A neighboring nodes. More specifically, each neighbor node, beginning with node B that was involved in the authentication phase, encrypts with its private key, the value that she receives from the previous neighbor node along with its certificate, adds his public key to the message and propagates it to another neighbor node. The first value that is received in this sequence is $nodeID_A$ and the final node in the sequence is node A that receives the sequence's final message. This final result of the encryption is the *securityID* provided to node A in order to generate the security DHT virtual space coordinates. Node A certificate is stored in these coordinates, after following the typical procedures for obtaining a zone [12]. An example of the above procedure, for a 2 dimensional virtual coordinate space DHT is presented in Figure 19.

For the generic form of the linking protocol, let $H = \left[HASH_i\,() \middle| i = \{1, d\} \right]$ be a set of hash functions that each one of them is responsible for generating one of the Security DHT *d* dimensional virtual space coordinates. Also, let $N = \left[N_i \middle| i = \{1, d\} \right]$ be a set of neighboring nodes of a candidate linking node N_0 and that $K = \left[(k_i^{pub}, k_i^{priv}) \middle| i = \{1, d\} \right]$ is the set of public – private key pairs (k_i^{pub}, k_i^{priv}) for each node N_i while $Cert = \left[cert_i \middle| i = \{1, d\} \right]$ is the set of certificates assigned to each node N_i. Assuming the *E()* is a public key encryption algorithm and that $nodeID_0$ is the identification number of the candidate linking node N_0, generation of N_0 node *securityID* and *d* dimensional virtual space coordinates of the security DHT can be done using the following proposed algorithm:

```
1. $S^{(0)} = nodeID_0$
2. For i = 1 to d
    a. Compute in $N_i$ :   $S^{(i)} = E\left[cert_i, S^{(i-1)}, ts\right]_{k_i^{priv}}, k_i^{pub}$
    b. Transmit $S^{(i)}$ to node $N_{i+1}$ or $N_0$ (when $N_{d+1}$ is reached)
3. $S^{(d)} = securityID$
4. $SecurityDHT\ coordinates = \left[ HASH_i\,(\mathrm{sec}urityID) \middle| i = \{1, d\} \right]$
```

The certificate is transmitted to the node that occupies the zone in the security DHT with the generated d-dimensional virtual coordinates (*Security DHT coordinates*) along with the certificate's nodeID and public key. The node in the security DHT verifies the validity of the certificate and stores it for future use.



P_1 = nodeID

$P_2 = E[cert_B, nodeID]_{B\ priv.key}$, B pub. key

$P_3 = E[cert_C, E[cert_B,nodeID]_{B\ priv.key}, B\ pub.\ key]_{C\ priv.\ key}$, C pub. Key

$P_4 = E[cert_D, E[cert_C, E[cert_B,nodeID]_{B\ priv.key}, B\ pub.\ key]_{C\ priv.\ key}, C\ pub.\ Key]_{D\ priv.\ key}$, D pub. key

P_5 : SecureID $=E[cert_E, E[cert_D, E[cert_C, E[cert_B,nodeID]_{B\ priv.key}, B\ pub.\ key]_{C\ priv.\ key}, C\ pub.\ Key]_{D\ priv.\ key}, D\ pub.\ key]_{E\ priv.\ key}$, E pub. key

Figure 19 - A linking phase example for 2 dimensional DHT.

10.1.6 Authentication through the p2p overlay

After linking, a new node is fully authenticated and can be considered part of the p2p overlay, As such, she can be involved in all services that the network provides and can act as an authenticator of other new nodes that can have a claim to its DHT zone.

However, to maintain the high security level and provide advanced services (e.x. accounting to its users), a live video streaming p2p system needs to constantly evaluate the authenticity of its nodes and update the authentication data accordingly. Considering the proposed structure described in the previous section, there are several operations that need to take place concerning node authenticity during context exchange within the p2p network. Those operations are the following:

- *Authentication between nodes.* Communication between nodes is performed through control message exchange and through context exchange. In the second case, a session between two nodes needs to be established and retained as long as the requested information are exchanged. After its initial authentication each node has to remain authenticated. So, during the session establishment the authenticity of the involved nodes needs to be verified using a mutual authentication protocol (Authentication Session protocol).

- *Certificate expiration and reissuing.* Each certificate provided by the CA is not valid for an indefinite amount of time since it has a strict expiration date. Therefore after a specific time interval, provided randomly by the CA during certificate generation, a node Certificate is useless and needs to be issued again. The Certificate expiration can be discovered during the Authentication session protocol.
- *SecurityID regeneration.* The *securityID* number is mechanism that is used in order to enhance the security of the overlay and prohibit malicious behavior from the node for which it is issued and the node acting as an authenticator. When, however, such behaviours are spotted in the network overlay, the *securityID* needs to be regenerated.

10.1.6.1 Authentication between nodes (Authentication session)

The session authentication protocol is executed before the data exchange between two nodes (node A_1 and A_2). Initially, a random number r_{A1} is generated by node A1, is singed along with its nodeID, public key and securityID using node A_1 private key and is send to the node A_2. Then, node A_2 performs the same action with node A_1 by signing with her private key a generated random number r_{A2}, the number r_{A1}, her nodeID, public key and securityID. Both nodes verify the digital signature they receive and the random number r_{A1}. Then, each node uses the securityID she has received to generate the security DHT zone virtual space coordinates where the certificates proving the authenticity of the nodes are stored. We refer to the nodes where the certificates are stored as authenticator nodes. So, using the p2p routing mechanisms with input the security DHT virtual space coordinates for each authenticator zone, those nodes can be reached by node A1 and node A2. A certify request is send to each authenticator node. Each request for certification consists of the nodeID and public key of node to be authenticated and of the node’s public key requesting the authentication. If the authenticator nodes return a positive verification reply then the authentication is completed successfully. Nodes A_1 and A_2 are authenticated mutually in parallel. Note, that the verification reply is encrypted using the public key of the node requesting the authentication information so as to avoid man in the middle attacks. The proposed Authentication session protocol is presented in detail in Figure 20.

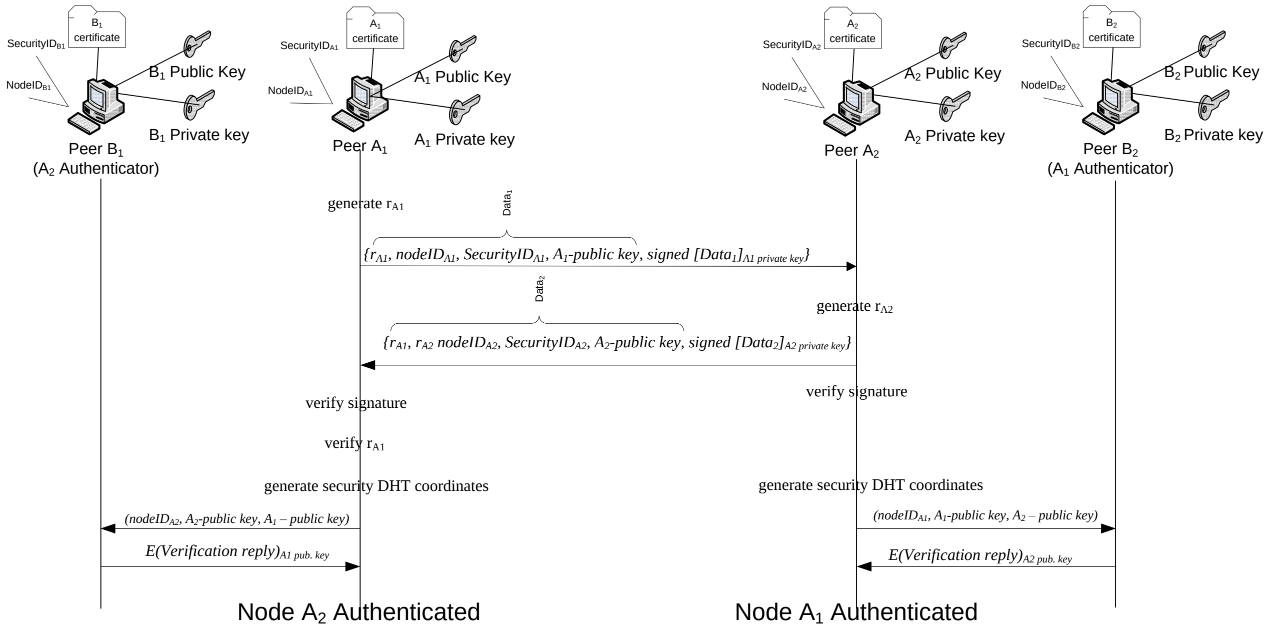

B$_1$ certificate
SecurityID$_{B1}$
NodeID$_{B1}$
B$_1$ Public Key
B$_1$ Private key
Peer B$_1$ (A$_2$ Authenticator)
A$_1$ certificate
SecurityID$_{A1}$
NodeID$_{A1}$
A$_1$ Public Key
A$_1$ Private key
Peer A$_1$
A$_2$ certificate
SecurityID$_{A2}$
NodeID$_{A2}$
A$_2$ Public Key
A$_2$ Private key
Peer A$_2$
B$_2$ certificate
SecurityID$_{B2}$
NodeID$_{B2}$
B$_2$ Public Key
B$_2$ Private key
Peer B$_2$ (A$_1$ Authenticator)
generate r$_{A1}$
Data$_1$
{r$_{A1}$, nodeID$_{A1}$, SecurityID$_{A1}$, A$_1$-public key, signed [Data$_1$]$_{A1\ private\ key}$}
Data$_2$
generate r$_{A2}$
{r$_{A1}$, r$_{A2}$ nodeID$_{A2}$, SecurityID$_{A2}$, A$_2$-public key, signed [Data$_2$]$_{A2\ private\ key}$}
verify signature
verify signature
verify r$_{A1}$
generate security DHT coordinates
generate security DHT coordinates
(nodeID$_{A2}$, A$_2$-public key, A$_1$ – public key)
(nodeID$_{A1}$, A$_1$-public key, A$_2$ – public key)
E(Verification reply)$_{A1\ pub.\ key}$
E(Verification reply)$_{A2\ pub.\ key}$
Node A$_2$ Authenticated
Node A$_1$ Authenticated

Figure 20 - The proposed Authentication protocol message exchange.

10.1.6.2 Certificate reissuing

When a certificate expires it needs to be reissued. However, in order to retain the scalability of the p2p network, the involvement of the CA in reissuing certificates should be minimized. Ideally, the CA should not get involved in the reissuing of a certificate after expiration as long as there is no other change in the certificate information. However, this is not possible since this policy would render the use of expiration dates in a certificate useless. In live video streaming p2p applications where video on demand scenarios are implemented, the issued certificates must be subject to reevaluation after some time intervals for authorization and accounting reasons. Therefore, certification expiration cannot be avoided. The alternative in retaining the network scalability is to minimizing, as least as possible, the intervention of the CA with the p2p nodes during certificate reissuing.

The expiration of a node certificate (node A) is revealed from an authenticator node during the authentication session protocol execution. In that case, it is the authenticator node's responsibility to communicate with the CA, acting as a representative of node A, use the nodeID and public key of node A and receive an updated certificate by the CA. However, this value is not useful for the authenticator node since it has the following form $E(\textit{updated certificate}, ts)_{\textit{nodeA pub.key}}$ where *E()* is a public key encryption function and *ts* is a timestamp value in order to protect against replay attacks. This value is transmitted through the node that requests node's A authentication to node A. Node A decrypts the certificate, verifies that is valid and using his *securityID* through the security DHT transmits the updated certificate to the authenticator node.

10.1.6.3 SecurityID regeneration

In all the securityID regeneration triggering events, the procedure that is followed remains the same. The *securityID* generation algorithm proposed in section VI is executed and a new securityID value is obtained. The node's (node A) old and new *securityID* are used in order to find the old and new certificate authenticator virtual space coordinates in the security DHT, respectively. Then, the certificate of node A is transmitted along with the nodeID and public key to the node (new authenticator) with a zone on the new virtual coordinates of the security DHT and a delete certificate request is transmitted to the old authenticator node.

10.1.7 Security Analysis

The proposed authentication scheme has to be resistant to a series of adversary attacks in order to be considered secure. Eavesdropping on the communication channel is a well known technique for mounting a series of traditional network attacks, like Man-in-the-Middle (MITM) or replay attacks. In the proposed scheme no sensitive information is transmitted unencrypted through the communication channel. Also, the use of random numbers and timestamps in the authentication phase during node addition on the DHT p2p overlay and the Authentication session protocol, prevent replay attacks. A MITM can not be mounted because data that can be exploited are encrypted. More specifically, the exploitation of the provided certificate by the CA can not be used by a potential adversary since it is encrypted by the requesting node's public key and thus can only be accessed by decryption using the node's private key (the private key is only known to the node that generated it). Data integrity is also maintained by the use of digital signature of the transmitted data. Digital signatures are also used as a proof of identity that is confirmed by the knowledge of the transmitted node's private key. (confirming that the claimed transmitted data from some node are truthfully send by this node). This approach was introduced in step 5 of the authentication phase in section 11.1.4.

However, the most potent danger for the p2p network security comes from Sybil and Eclipse attacks. The proposed scheme's protection measures against such attacks are structured based on the principle that conspiracies between neighbor nodes should be very difficult and that if such conspiracy is discovered then the involved nodes cannot deny this action and can be easily removed from the p2p overlay. To achieve this we enforce a strict authentication mechanism based on public key certificates with random expiration date. The CA is the only one responsible for issuing such certificates and requires only the node's public key and IP address to do that. The node identity (nodeID) is provided by the CA and cannot be denied since such action would result in immediate castoff from the network's DHT (as dictated by the

Authentication session protocol). Also, to prevent the node from choosing her place in the DHT, the node's zone d-dimensional virtual space coordinates are provided by the CA and are stored in the issued certificate. Thus, the new node does not know where she will be placed in the DHT. Even if a user manages to insert multiple identities – nodes in the DHT he has no knowledge of the zones where those nodes will be placed. This prevents the cooperation of such malicious nodes and renders p2p related attacks very difficult.

To protect the certificate integrity and further enhance trust between nodes, we introduce the notion of a security DHT. Using this approach, the certificate of a node after generation and addition in the DHT is not only stored in the certified node but also in a randomly selected node that acts as an authentication advocate. Authentication is handled by this authenticator node and therefore the certificate cannot be changed or manipulated even if the node for which it was issued wants to do that. The coordinates of the authenticator node in the security DHT are provided by a random number, the *securityID*, that is provided by the certificated node's neighbors. Since the neighbor nodes are placed randomly in the DHT they cannot be part of a conspiracy. Thus, the *securityID* value can be considered random. Its authenticity, if required, can be verified by reversing the *securityID* generation algorithm described in section 11.1.3.

10.1.8 Evaluation

For the network performance of our system we have used Opnet modeler v14 and we have developed a packet level simulator. We implement our system by using CAN as a service overlay in order to evaluate the latency of our p2p authentication and the network bandwidth that is consumed. In our system we consider a scenario where 2000 peers entered the overlay and authenticated in accordance to our proposed scheme In order to have authentication through our proposed system there is the need to route a message through the DHT in order to find the peer that is responsible to perform the authentication. Towards this goal in Figure 21 we performed 2000 authentications and we demonstrate the cumulative density function of the network stretch that our system introduces. Network stretch is defined as the ratio between the latency of our system, for the routing of a message through the DHT, and the latency of the direct network path for the transmission of a network packet. As we observe the network stretch is on average around 4 in case of a DHT agnostic to the underlying network and less than 2 on average in case of a DHT that is adapted to the underlying network.

In order to evaluate the additional network bandwidth that is consumed for each authentication when we compare our system with a conventional centralized authentication scheme we have to add the network bandwidth that consumed for the routing through the DHT and the transmission of the peers zone. The former is the routing through the DHT that requires a number of hops that are logarithmic with base equal to the number of the DHT dimensions. So in a DHT where N peers participate we need $\log_d N$ hops on average for reaching the responsible peer. The zone of each peer has length that is less than 8 bytes and it is based on the dimensions of the DHT.

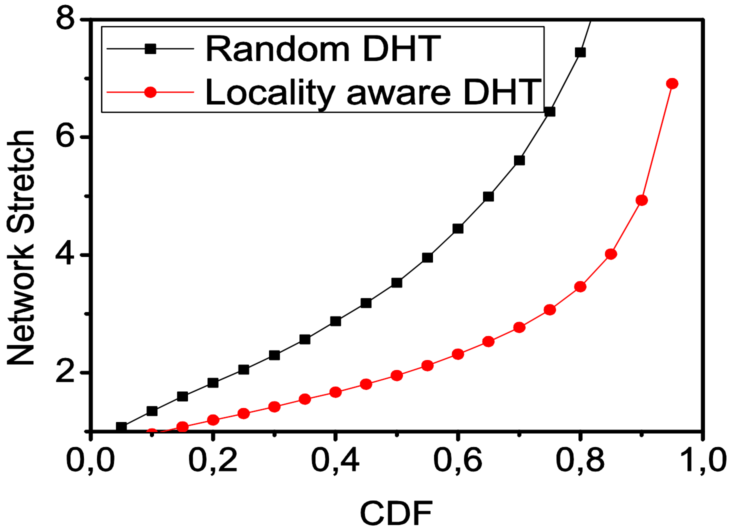


Figure 21 - Network stretch of CAN and a locality aware CAN .

Performance results on the total authentication scheme can be taken by using 2048 bit RSA encryption – decryption and keys. Therefore, a public or private key has a length of 256 bytes plus 256 bytes for the RSA Modulus, the nodeID has a length of 32 bytes, the securityID is of 256 bytes. A certificate consists of the nodeID, the public key, the public modulus, the IP address, the expiration date (2 bytes) and the node’s zone coordinates.

11 Description of accounting system

Accounting in Vital++ attempts to blend IMS accounting specifications for pre-pay and post-pay with P2P accounting mechanisms to incentivise good network behaviour. For example a Vital++ network may offer pricing discounts to subscribers who participate in overlays, because they provide content to other subscribers. Rather than implementing an accounting system specifically catered for a particular accounting formula, we have designed a system that can handle an arbitrarily complex charging scheme taking data from the network, overlay management system and content protection subsystem.

The Accounting Subsystem is not intended to replace an operator's existing charging and billing infrastructure. Instead, it develops a content-charging model where charging data, rating schemes and resulting subscriber bills are generated in the context of content licensing by its own. The CREST system described within this deliverable is triggered by the Content Protection Subsystem described within D4.3.

The Accounting Subsystem provides interfaces for Content Providers to associate a charging scheme with their content. It also exposes web service and diameter interfaces to receive network and content based charging information. The actual rating is carried out using the Internet Protocol Detail Records. Rating schemes are spreadsheet worksheets, which can support arbitrarily complex rating schemes incorporating conditional rules.

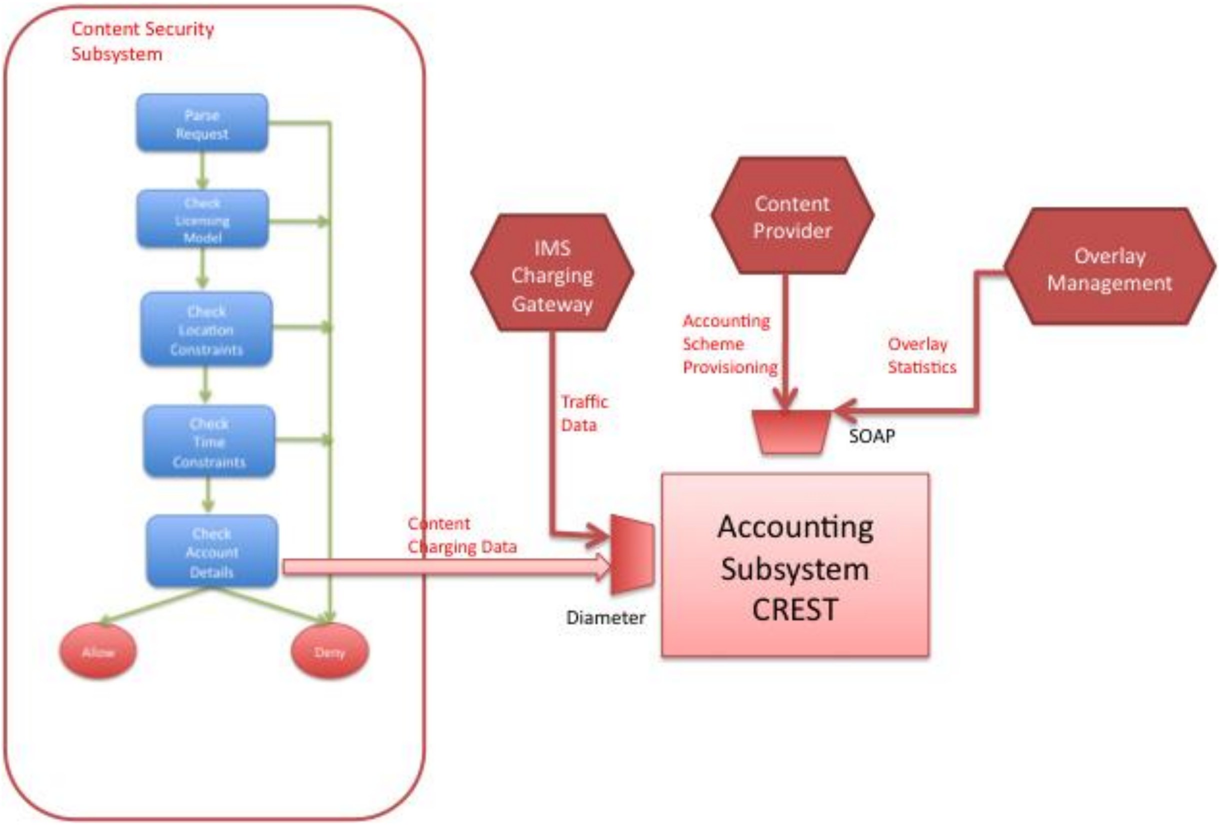


Figure 22 - Accounting System Interfaces

11.1 Authentication for Accounting

As described, the Accounting Subsystem is triggered by the licensing process of the Content Protection Subsystem described in D4.3. This subsystem reuses the PKI Certificate Authority of the P2P authentication mechanism described in this deliverable in order to:

- Mutually authenticate Content Provider with Content Consumer;
- Ensure non-repudiation in the licensing process;
- Encrypt the symmetric (AES) key used to “super-distribute” the data. This is further described in D4.3.

11.2 CREST Rating Engine

To meet the demands of the Vital++ service scenarios and their respective accounting needs, the CREST rating engine has been chosen and further developed to match Vital++ specific requirements. CREST is a flexible, component-oriented rating engine, which is easily configurable and extensible. Thus, the CREST engine meets the needs of the project.

The core rating functionality of CREST is based on the concept of encapsulating a charging scheme for a service in a spreadsheet. All logic required for calculating the scheme can easily be contained in a spreadsheet. Furthermore, the syntax of a spreadsheet itself tends to easy manipulation by non-programmers. In this way, charging algorithms can quickly be modified and/or created by someone else, without knowledge of a high level language or an understanding on how the system works.

CREST also utilises IPDR as the underlying usage data record format, although it can be extended to cater for other record formats, too. CREST can be deployed as a scalable, distributed, asynchronous, message-driven solution or as a compact, single seat deployment with synchronous calls for on-demand rating queries. In this way, CREST is ideally suited to meet the needs of Vital++.

11.3CREST Architecture

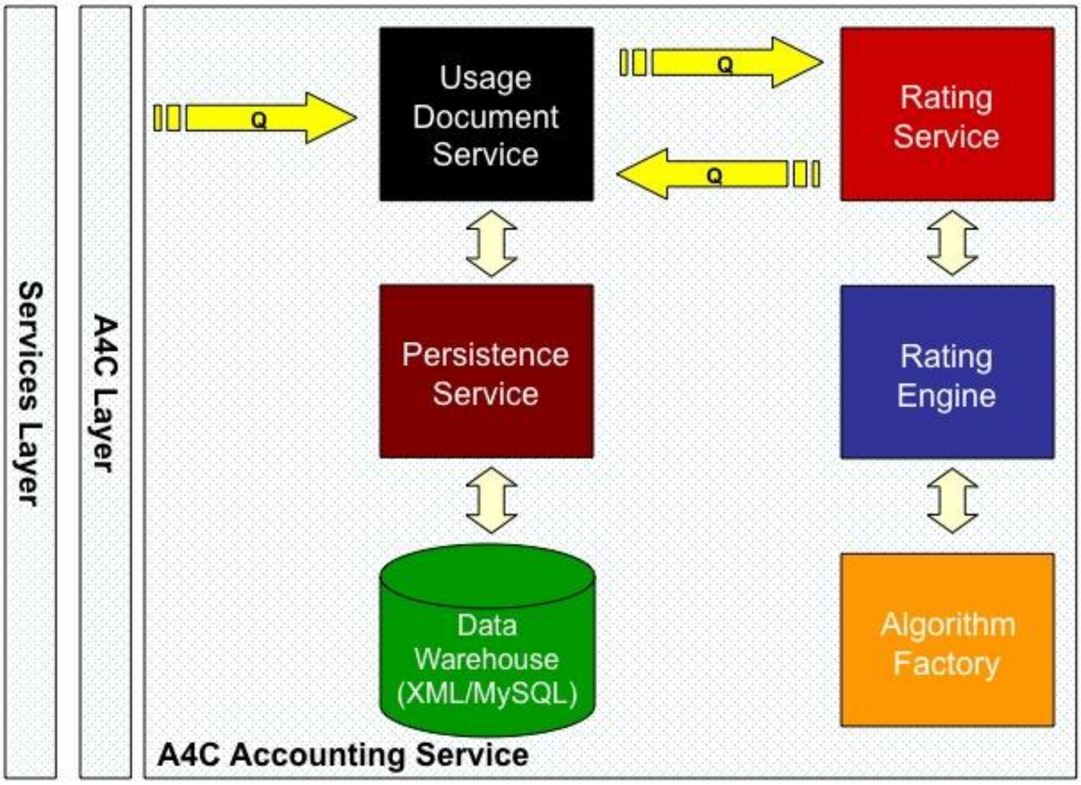


Figure 23 - CREST Architecture

As seen in Figure 23, CREST comprises a number of standalone components that interact to form the service accounting solution. Many of these components are easily exchangeable for alternative implementations but the architecture remains the same. CREST can be broken down into 4 main subsystems: Usage Document Subsystem, Queuing Subsystem, Rating Subsystem, and Persistence Subsystem; each of which deals with a specific aspect of the rating process.

11.3.1 Usage Document Subsystem

The Usage Document Subsystem of CREST is currently implemented to handle IPDR documents passed to it. It initially maintains the usage document without changing it (for the purposes of auditing) and then passes the usage document to the Rating Service for processing. This component of the system can be configured to run as a message-driven queue implementation, which makes it useful for managing large volumes of records being submitted for processing.

11.3.2 Queuing Subsystem

The Queuing Subsystem of CREST is an abstracted implementation of the OpenJMS queuing specification, utilising the OpenJMS queue server. This service allows for various components of CREST to be wired together to act in a message-driven environment should it be required.

11.3.3 Rating Subsystem

The Rating Subsystem of CREST is by far the most complex component of the overall system. It incorporates various engines that combine the usage documents in order to process them and to produce an associated charge for the usage instance passed to it. The core of the component is the algorithm engine, which uses a programmatic abstraction of a spreadsheet to load, and process algorithms contained in workbooks and to pass various usage data parameters to the algorithm as required. Once the algorithm has completed its job and the final charge for the usage instance has been calculated it is then extracted and inserted into a Charge Element within the IPDR, along with details of the algorithm used to rate the service usage, before finally being persisted for subsequent processors such as a billing engine to query.

11.3.4 Persistence Subsystem

The Persistence Subsystem of CREST is a database abstraction layer used to aid the persistence of IPDRs passed into the system. For the purposes of Vital++ this layer will utilise an underlying MySQL implementation so that IPDRs can be serialised to the database. However, sibling implementations also exist for native XML databases such as eXist and Berkeley DB, which can be activated as required through the system configuration files.

11.4Operation of the Components

Below is an example of the CREST in operation for a typical rating scenario. The sequence chart depicts the message flow from system initialisation to usage instance rating. For clarity, persistence calls have been omitted from the diagram. However, it can be assumed that the usage document is persisted to a chosen database both before and after rating.

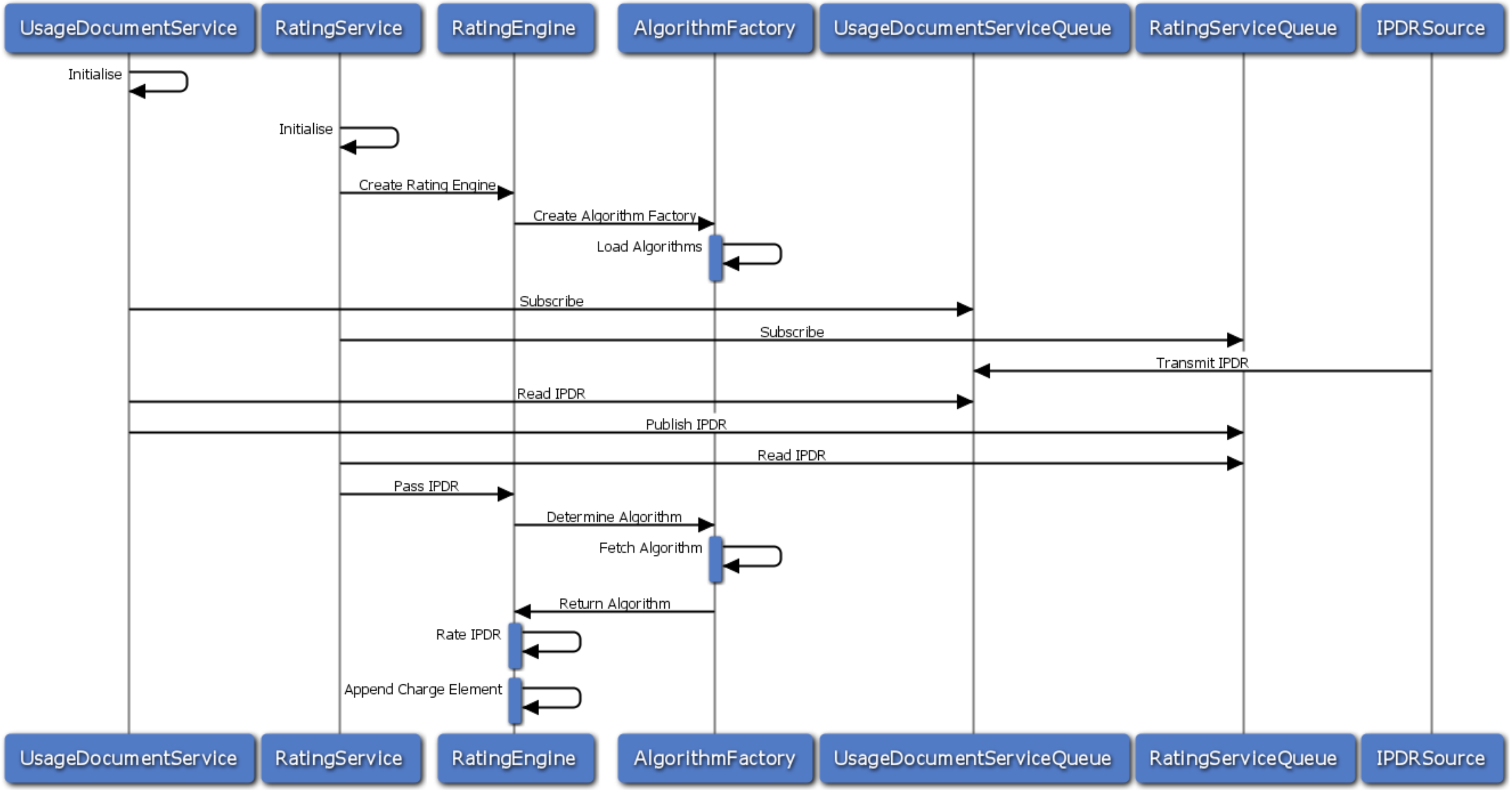


Figure 24 - CREST Rating Scenario Sequence Diagram

11.5CREST Interface

There are a number of ways for developers to interface with the CREST system. Some subsystems are optional, although recommended for full deployment, e.g. the Usage Document Subsystem can be bypassed in favour of communicating directly with the Rating Subsystem without any effect on the core functionality of CREST.

A typical scenario for a full deployment of CREST involves a developer, who is creating a metering agent for a service, which in turn produces IPDR documents. This metering agent can also be implemented as a queue publisher for the Usage Document Service Queue. Every IPDR produced is published to the queue and the results can later be retrieved from the database through a billing engine or other, preferred technique. Interfacing with the Usage Document Service Queue is currently performed through a native Java API but this can be easily wrapped in another interface layer to provide maximum connectivity between service providers and the CREST system.

Developers, who want to bypass the Usage Document Subsystem with the intention to directly interact with the Rating Subsystem, have two options: 1) Pure Java API, 2) Web Service API.

1) The pure Java API for the Rating Subsystem allows developers to pass IPDR documents directly to the rating engine for processing. Developers can use asynchronous, hands off, rating with subsequent retrieval of the rated documents through the billing/other interface. Alternatively, they can avail of a synchronous call that will return the rated document upon completion, which might be useful for testing/retrieving quotes for service usage.
2) The Web Service API provides developers with an implementation independent means of communicating with the Rating Subsystem. Facilitating the passage of IPDR documents to the rating engine and subsequent return of a rated IPDR instance, this web service API provides a synchronous means of communicating with the Rating Subsystem and provides maximum flexibility for service providers in terms of interoperability. Returning the rated IPDR instance gives the service provider the choice of utilising a billing system built atop CREST or alternatively using their own data store for rated instance persistence and retrieval.

11.6Billing Interface

To provide service users with a mechanism of assessing their service usage, a billing interface will be created for the CREST. This interface will consist of an easy to use website that is personalised to the end user, post login. Following the standard procedures of telco service providers, the website will list the service subscriptions of the users logged in and provide them their latest bill. The bill itself will be viewable online or downloadable in a variety of formats for maximum compatibility with the end-user’s computer software.

Hooks used by the website to garner pertinent information for the purposes of bill creation and user addressing, will also be documented as a high-level Java API. This provides that future development could investigate the possibility of exposing the API in a secure fashion so that service providers may create their own customised billing solutions rather than availing of the hosted solution.

12 Integration with other architectural components

Authentication, authorization and accounting components and procedures implemented in the context of Vital++ are integrated with other subsystems residing either on the client side or on the Application Server side. The foreseen integration is performed at runtime as a sequence of interactions according to a number of principles based on elimination of user anonymity and content protection.

User Authentication itself is based on the use of Digital Certificates in the context of the P2PA-SA operation. It is integrated with the client components so as to enable verification of the identity of the communicating parties. SIP based messaging ensures that the IMS infrastructure authenticates the messages against the relevant account prior to forwarding the message to the recipient. Nevertheless, there are messages which are transmitted directly and bypass the IMS procedures via existing overlays or DHT mechanisms in this way. They require a verification action to be performed by the user agent. In this case, the UA has to retrieve the certificate of the other party by invoking the "P2PA RequestCert" process in case no formerly exchanged certificates have been cached.

Authorization of users is closely related to content discovery and acquisition. In the process of discovery the foreseen integration is based on the fact that CI-SA checks with the CP-SA if a specific content item that is already published is allowed to be listed in the response to a user's query. This permission checking is performed by the CP-SA on the basis of the user identity against the stored business rules for the specific publication. If the specific item is not clarified as displayable to the specific user it is removed from the result list.

If a user is authorised for a specific item listed in a query response and if there is a specific request for acquiring it, the client side has to interact with the CP-SA in order to acquire a license for it. At this point the authorization entity, i.e. the CP-SA, has to contact the Accounting service in order to provide charging information or, in the case of pre-paid, to clarify if the licensing procedure can be completed or not.

Finally, the CI-SA checks with the CP-SA, if there has been a successful licensing procedure for the requested item.

13 Conclusions

In this document we have described and analyzed the Vital++ functionalities for authentication and accounting after introducing the most common security risks in a P2P network.. Authentication in this scope refers to the authentication between arbitrary Vital++ nodes, i.e. mainly clients but also central components like the Content Protection Subsystem (CPS). The presented mechanism for P2P-Authentication uses a common certificate authority to subscribe peer-generated certificates, which identify the corresponding peer in a secure way. These Certificates can be used to proof the own identity to other peers, but can also be used to verify signatures of critical P2P-message, e.g. during overlay (re-)organisation or instant messages, etc.

Further development of the introduced P2P-Authentication mechanism can improve the overall availability, performance and scalability by using a secure distributed hash table (DHT) for storing relevant Vital++ information, like peer contact addresses, certificates, profiles, buddy lists, etc.

P2P-Authentication also enables the establishment of encrypted P2P channels by applying the Diffie-Hellman key-agreement algorithm in order to generate symmetric keys, e.g. for the Advanced Encryption Standard (AES) algorithm.

This mechanism is being used e.g. CPS, as license objects need to be transmitted only in encrypted form in order to keep media crypto keys secret from potential eavesdroppers.

Vital++'s DRM system took its requirements from real content provider requirements and so is related to real-world business requirements. In more detail the requirements are: Identity based Conditional Access to Content, flexible rights expression, Accounting, privacy.

The accounting solution of Vital++ is a subcomponent of the CPS and provides APIs for content providers in order to specify arbitrarily complex accounting and charging schemes. The CREST rating engine is the core component of this functional block. The accounting subsystem receives overlay statistics from the overlay management subsystem in order to achieve information about peer activity. It also uses the standardized IMS interfaces for charging and billing. The Accounting subsystem is being triggered by the CPS, e.g. when a user requests a license, which is related to a content object, which is related to an accounting scheme.

Thus the requirements are met, as the accounting subsystem together with the P2P-Authentication and the CPS will provide the related features (s.a.).

Conclusively, the problem of P2P-trust will always require a common trusted authority, which in the scope of this project is being realized as an application server in the IMS. The developed accounting system is capable of rating charging data to produce subscriber bills based on content pricing and overlay network traffic statistics. However, we are again reliant, on a centralized application server in the IMS, the Overlay Manager in this case, to verify these overlay statistics. We believe that this makes sense in the context of a typical IMS deployment where the network operator will wish to retain control of accounting functions.

14 References

[1] IETF Network Working Group, RFC 3310, World Wide Web, http://www.ietf.org/rfc/rfc3310.txt

[2] TeleManagement Forum. Telecom Operations Map, approved version 2.1 edition, March 2000.

[3] IETF AAA Working Group. RFC 2975. World Wide Web, http://www.ietf.org/rfc/rfc2975.txt, November 2000.

[4] A. Heintz and Dr. M. Lucas. IP Pricing and Rating. Billing World, June 1999.

[5] ipdr.org. Network Data Management Usage (NDM-U) for IP-based Services, version 3.1.1 edition, October 2002.

[6] ITU-T, ISO/IEC. Abstract syntax notation one (asn.1), specification of basic notation, itu-t rec. x.680 (2002) | iso/iec 8824-1:2002, 2002.

[7] Thomas J. McCabe and Arthur H. Watson. Software complexity. Crosstalk, Journal of Defense Software Engineering, December 1994.

[8] A. Pras, Bert-Jan van Beijnum, Ron Sprenkels, and Robert Parhonyi. Internet accounting. IEEE Communications Magazine, vol. 39. no. 5, May 2001.

[9] Openet Telecom. Fusionworks activerate product brochure, May 2005.

[10] G. Zhang, B. Reuther, P. Mueller. User orientated ip accounting in multi-user systems. In INM 2003, pages 59-72. IEEE, 2003.

[11] Fabio Picconi and Laurent Massoulie, Is there a future for mesh-based live video streaming? IEEE P2P 2008

[12] S. Ratnasamy *et al.*, A Scalable Content Addressable Network, Proc. ACM SIGCOMM, 2001

[13] S. Čapkun, L. Buttyán, and J. Hubaux, 2003. “Self-Organized Public-Key Management for Mobile Ad Hoc Networks”. *IEEE Transactions on Mobile Computing* vo2, issue 1, pp. 52-64, Jan. 2003.

[14] Guido Urdaneta, Guillaume Pierre and Maarten van Steen. “A Survey of DHT Security Techniques”, *ACM Computing Surveys*, 2009.

[15] Dan S. Wallach, " A Survey of Peer-to-Peer Security Issues",Software Security -Theories and Systems, LNCS 2003, pp 253-258, Springer Berlin,

2003.

[16] J. R. Douceur, « The Sybil Attack” in *Revised Papers From the First international Workshop on Peer-To-Peer Systems* (March 07 - 08, 2002), editors: P. Druschel, M. F. Kaashoek, and A. I. Rowstron, LNCS vol. 2429, pp. 251-260, Springer-Verlag, London, 2002.

[17] E. Sit, and R. Morris, “Security Considerations for Peer-to-Peer Distributed Hash Tables”, in *Revised Papers From the First international Workshop on Peer-To-Peer Systems,* Eds: P. Druschel, M. F. Kaashoek, and A. I. Rowstron, LNCS vol. 2429, Springer-Verlag, pp 261-269, March 07 - 08, 2002

[18] C. Lesniewski-Laas, “A Sybil-proof one-hop DHT”, In *Proceedings of the 1st Workshop on Social Network Systems* (SocialNets '08). pp 19-24, ACM, New York, NY,. Glasgow, Scotland, April 01, 2008.

[19] P. Tsang and S. Smith, “Ppaa: Peer-to-peer anonymous authentication,”, in proceedings of 6th International Conference on Applied Cryptography and Network Security (ACNS 2008), pp. 55–74, New York, NY, USA, June 3-6, 2008.

[20] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, “Secure routing for structured peer-to-peer overlay networks”. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation* (OSDI '02), ACM, New York, NY, pp. 299-314, Boston, Massachusetts, December 09 - 11, 2002 .

[21] Nikolaos Efthymiopoulos, Athanasios Christakidis, Spyros Denazis, Odysseas Koufopavlou L-CAN Locality aware structured overlay for P2P live streaming, 11th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services (MMNS) 22-26 September 2008, Samos, Greece

[22] L. Zhou and Z. Haas, “Securing Ad Hoc Networks,” *IEEE Network*, vol. 13, no. 6, pp. 24–30, November/December 1999

[23] A. Boukerch, L. Xu, and K. EL-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer Communications*, 30:2413–2427, September 2007

- End of document -