A Public Keys Based Model for P2P Content Authenticity, Access Control and Reputation

Neander Larsen Brisola, Altair Olivo Santin, Lau Cheuk Lung, Heverson Borba Ribeiro
Graduate Program on Computer Science – PPGIa / Pontifical Catholic University of Paraná – PUCPR
Curitiba – Paraná – Brazil
{neander, santin, lau, heverson}@ppgia.pucpr.br

Abstract — In the classic use of P2P (file sharing, mainly, music and movies) there is not the same concern with authenticity and content access control. Security proposals currently found in technical literature attempt to adapt techniques of client-server architecture to the P2P environment, which does not seem to be the most appropriate approach. This work proposes the usage of a more flexible, secure and appropriate approach to the P2P environment, the SDKI/SPKI (Simple Distributed Security Infrastructure/Simple Public Key Infrastructure). It is shown in the proposal that the use of public keys to identify peer allows the creation of a persistent identification scheme, without loosing anonymity, even in a self-managed environment as P2P. In addition, the usage of digital signature to provide authenticity to P2P content is adopted. In order to provide credibility to the public keys used by the SDSI/SPKI a reputation based approach is applied. A scheme is also proposed in order to guarantee non-repudiation in the transfer of P2P contents. The implementation of a prototype showed that the propagation of polluted content can be reduced from the current 60% up to 80% to zero percent in cases of the proposed scheme.

Keywords: P2P security, authentication, authorization, reputation

I. INTRODUCTION

Peer-to-Peer networks (P2P) allow end-to-end communication regardless of the underlying network (in general networks based on *Internet Protocol*). That is, physical limits imposed by networks based on IPs are transposed and the internet becomes supportive to an unlimited overlay network – a web of P2P networks operating on an IP network (for instance, gnutella, eDonkey, freenet, BitTorrent etc.).

Currently, P2P networks are intended to be non-structured (decentralized), presenting themselves as an alternative to client-server architecture, in this case, a peer (nodes of P2P network) can be server or client (a *servant*) – depending on direction of content flow. P2P has the advantage of scalability, fault tolerance and an increasing number of available resources; for many the support of anonymity is also a peer-to-peer advantage. P2P is an excellent alternative for computer infrastructure cost reduction, because tasks may be distributed to peers avoiding the purchase of servers (specific hardware and software), for example. The servant has more and more available resources (processor, memory and disk) in addition to the network connection with greater and greater available bandwidth.

One of the P2P features according to Dingledine and his colleagues [1] is the support to anonymity that may be of several types; anonymity of: document, reader, server, author, publishing and search. In all cases the objective is to prevent a third party from being able to find out, respectively: stored documents in a peer, who accessed the document, which peer stores which document, which user created which document, which publishing peer published which document and that no one other than the server knows which document is being searched for.

There are risks involved in the use of P2P networks, intentional or accidental sharing of sensible information can cause serious damages. Furthermore, a servant, client P2P, is also a server being executed in a machine of a user. This server can be attacked as well as any server currently in use on the internet, and exploited using a buffer overflow, for example. A programming error in a servant exposes the host to intruders. However, if the content is well stored (distributed and encrypted, for example), the action of an intruder will be made difficult, contrary to what happens in a client-server architecture, for example, where server vulnerability exposes the whole system to high risk. Generally, also, there are no security weakness alerts in P2P software or vulnerabilities patches released, what are normally issued are only new versions reporting software enhancements.

In general in P2P networks there is no concern with authenticity of content, i.e., any user can modify content and publish it with the same original description (keywords). A user searching for certain content would find the authentic and modified content through keywords and would be confused, because they would have to choose one of the results of the search by chance. Actually, if they choose the modified content, the user would not even know that the original had been modified and published with the same description. Therefore, in addition to being misled the user would share this non-authentic content (polluted) with other network users and would become a passive replicator of P2P junk [2].

An even worse situation can happen when a peer publishes an attractive description (through keywords), however, which have nothing to do with the shared content. A user searching for content with those keywords could waste time and network bandwidth downloading a content which is actually fake (P2P junk). In 2005 the percentage of polluted

content already represented 50% of P2P traffic which in some providers already implies 60% of the total internet traffic [2].

In order to solve the aforementioned problem of lack of authenticity, approaches based in authority, voting and reputation may be considered [3]. Approach base on authority takes on a node (authority) which signs all documents created by peer users, the signature verification gives the content authenticity [4] [5]. However, some authentic signatures might have invalid content.

In current P2P networks in use and technical literature, authenticity of content in general is obtained using digital signature without certification. Due to dynamism and the features of P2P networks, normally, there are no signatures of certifying authorities in the certificates, which means that certificates are auto-signed [6]. In practice, auto-signed certificates are suitable only in order for protection per message at communication channel level in the sending of encrypted contents with SSL/TLS, for example [4] [5].

The disadvantage of auto-signed certificates is that there is no well-known entity in order to provide the endorsement of certificate, as a PKI certifying authority, for example. In this case, nodes with positive reputation in previous access can be considered to give credibility to peers sending auto-signed certificates. The problem of this approach is that in practice peers constantly change identification in order to support anonymity [7] [8]. Therefore, it is necessary to create persistent mechanisms of identification for peers, considering that the network is driven by content and not a client-server architecture.

In any system involving security it is worth providing resources which prevent false denial of participation in a content exchange to support non-repudiation. In the case of networks driven by content (P2P) this must be accomplished without compromising anonymity [1].

Access control in P2P networks does not seem very applicable seeing as these networks were designed for free content sharing, however, if the P2P network is used as a distributed infrastructure for corporate and academic use, for instance, such control is required. There are several proposals of access control in the literature [9] [10], however, most of them use auto-signed certificates for authentication of source and apply classic access control (descriptive by ACL) based on TTP (Trusted Third Party) on destination. Others use RBAC mechanisms for access control [11], i.e. all proposals try to support classic access control of client-server architecture in P2P network.

Classic access control depends on a server where the control imposition is effected (by enforcement mechanism). As in P2P network the content is distributed over the servants, whether there is not a server for content there is not a path (URI) where the content can be found, therefore how to control access to content in this case?

SDSI/SPKI [12] was created to support access control and developing of secure distributed systems. SPKI uses an equalitarian model, without entities centralizing/concentrating authority. SPKI certificates are auto-signed and support anonymity. Access control is based on keys and the rights are encoded in certificates propagated through authorization chains [13]. Authenticity mechanism is digital signature. One of the difficulties in the SPKI is the storage of certificates and search of authorization chain, when the principal does not participate in it [14].

In this work we propose a scheme for identification of P2P nodes, support to anonymity, and assuring non-repudiation to P2P nodes and credibility to SDSI/SPKI keys. SDSI/SPKI keys will be used to generate digital signatures which will assure the authenticity of P2P content. It also proposes the usage of a repository for storage and for search of SPKI certificates. In addition, it uses SPKI chain certificates for access control to P2P contents.

This work is structured in the following way, in section 2 peer-to-peer technologies are presented (JXTA and DHT), section 3 describes SPKI/SDSKI, section 4 specifies details of proposal. Section 5 provides related work. Section 6 illustrates Proposal Considerations and section 7 draws a Conclusion.

II. PEER-TO-PEER NETWORKS (P2P)

P2P networks show excellent potential as infrastructure/middleware [15] for storage, search and sharing of contents in distributed environment. In addition, they are easily scalable, fault tolerant, decentralized and currently amply broadcast.

In the beginning P2P network became known for file sharing through Napster[1], also because of it P2P networks are synonymous with file sharing until today. However, many other uses can be applied to a P2P network, for example: Content distribution; instant messages; IP telephony; distributed processing etc.

Corporate use of P2P networks can additionally benefit the company immensely, especially, in activities where there is a need for cooperation in projects, customer relationship management systems, file storage, remote backup, instant messaging services etc. A study carried out in 2003 by Frost & Sullivan [16] estimated the number of corporate users of P2P networks, in the United States, in that year was 100,000 and for 2006 it is estimated at 1.8 billion users.

There are several ways of implementing P2P networks, however basically, there are two models which are different regarding connection control: brokered and pure. In the first case, regarding nodes, which search for certain, content, to connect to a server node, need beforehand to receive indications from a central node (super node). In this model, connection control is client-server, after the client obtains information of servants; since the connection is brokered by the super node this model is known as the “hybrid” model [17].

The example most known of brokered control was the file sharing network Napster; General features such as indexing and searches are performed in the intermediate server located

[1] http://www.napster.com/

in the P2P network. Other examples of this network type are Groove Network, Kazaa and Blubster.

In the pure model, nodes communicate between themselves through a direct connection for resource sharing as well as obtaining location information; this makes the P2P network architecture more flexible. In this approach nodes shall have control over their own organization, routing, and other control features for management of P2P networks. There is no centralized node to mediate connections in P2P networks. Gnutella[2] is a popular example which employs feature control this way. In the Gnutella network, each node manages relevance and searches forming P2P networks, the connection is made through messages used in the search flooding technique to neighbor nodes. Some known examples of this model of P2P networks, denominated as Morpheus, Limewire, FreeNet, JXTA, and Publius.

A. Indexing P2P content

By its distributed nature, P2P depends greatly on indexing services to facilitate content search; various strategies were adopted for it. The simplest strategy, without index is based on the technique of search per flooding and cache of results. Therefore, afterw the first search of the same subject the cache becomes the search index; the other most efficient search technique is the usage of DHT (Distributed Hash Table).

DHT consists of a hash table implemented in a distributed manner, under some form of structuring (ring, tree etc). It is called hash table because all data stored in the table pass through a hash function (MD-5 or SHA-1) before being inserted in the table [18][19]. After this procedure an ordered pair <key,value> which is stored in the tables of the DHT is created. Nodes which store DHT data, keep some additional (routing) information on other nodes to facilitate their location.

In order to perform searches on node of DHT the peer bases on information about known nodes and also about the peers which are near to the target peer (that are intended to be reached by the search). Through the search it is possible to recover a certain value from a key provided in a search on DHT. Some examples of DHTs that emerged in 2001 are CAN, Chord, Pastry and Tapestry and, since then, it has being multiplied in innumerous approaches and distinct implementations.

DHT is scalable, fault tolerant, deterministic in the search and can be entirely built over preexistent technology.

B. JXTA

Jxta [20] is a set of protocols based on XML created in order to supply P2P networks typical functionalities. Its approach is independent of platform or language, offering architecture for creation of P2P application.

Jxta creates a logical layer over the physical layer, decreasing the communication complexity between devices of heterogeneous networks. This way Jxta protocols establish an overlay network over the Internet, allowing interaction of nodes regardless of their location; Jxta transposes Firewalls.

As identifier Jxta applies UUID, a 128-bit datum to refer an entity (a peer, an advertisement, a service, etc.). Once a peer gets an UUID, its can communicate with other peers through the Jxta protocols; it is possible to find advertisements, peers, peer group, and so on.

Jxta applies TLS (*Secure Transport Layer)*, based on PKI X.509 technology, that is suitable to provide protection to communication at messaging level.

III. SIMPLE DISTRIBUTED SECURITY INFRASTRUCTURE (SDSI) / SIMPLE PUBLIC KEY INFRASTRUCTURE (SPKI)

SDSI / SPKI is a simple PKI, based in authorization for distributed applications. SDSI/SPKI is client oriented and does not need server infrastructure for its operation. SDSI/SPKI is totally decentralized and technologically independent, allowing the storage of certificates in any type of repository [21][22]. SPKI is guided by authorization, meaning that the use of certificates – that are broadcast across the network – do not require servers for ACL storage. Additionally, account registration for users to access a server resource is not required; it is enough to have a public key and a authorization chain [23]. SDSI/SPKI supports anonymity through the usage of a public key for principal identification. SDSI/SPKI guaranteeing authenticity based on digital signature. Furthermore, it may be utilized to avoid non-repudiation, since all the exchanges of messages need to be digitally signed.

SPKI/SDSI has two types of certificate, names and authorization. The name certificates associate SDSI names to the public key or other SDSI names. The naming system is adopted by SDSI that induces the use of local names in the sense of a globally distributed environment. The SDSI names are always local, corresponding to the space names of the issued certificate. The principal issuer of the certificate is always identified by a public key. The public key combination rather local name forms a unique global identifier [24].

An egalitarian model is used in SPKI/SDSI; the principals are public key that may sign and publish certificates, such as CA of X.509. Therefore, any principal may create a pair of keys (private and public) and then associate the public key to a name in its local space of names and divulge them through certificates. The need of a centralized entity that registers the public keys and issues certificates such as CA of PKI X.509 is excluded. Thus, each principal defines the way that appears more intuitive, in its space of names, the names to other principals.

Through the authorization certificate the principal (issuer) delegates access permissions to other principals (subjects) in the system. The propagation of rights from issuer (server) to subject (client) create an authorization chain (sequence of certificates) and therefore a trust path between issuer and subject. When accessing a resource protected by the SDSI/SPKI scheme the client needs to present to the server a chain of certificates granting access to document (object) along with the signed request to do that.

The server verifies authenticity of chain and whether the delegated rights are suitable to get access to an object. To verify whether the access is granted to a client presenting a

[2] http://www.gnutella.com/

chain, the server verifies the signature on the request obtaining the last public key of the chain (that must be the client's public key) [25].

IV. THE PROPOSED MODEL

The greatest restriction for more intensive P2P network use for the purpose of, not merely sharing files (mainly music and film), in a corporate or commercial environment is without doubt relative to security aspects. Having risks involved in the professional use of P2P networks, the sharing (intentional or accidental) of sensitive information of a company, for example, could cause them incalculable damages. The proposal intends to cover security aspects that until nowadays have not been related by any other work, with a PKI infrastructure that proposes to be more adequate for self-management, decentralization, scalability, and flexibility of P2P networks.

Aiming at abstracting the security layer from application in this work, an intermediate layer between the application layer and the P2P infrastructure is proposed. The proposal has the objective of assuring, at application level, some security properties such as authenticity, integrity, non-repudiation, and confidentiality in the sharing of content in P2P networks. Figure 1 shows the general architecture of the proposed model for each peer network.

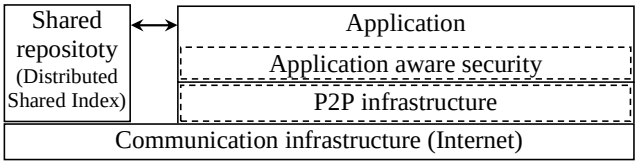

Figure 1. General Architecture of the Proposed Model

Software developed for use in P2P infrastructure can be found in the application layer. Interposing the application layer and the P2P infrastructure is a security layer which is not transparent for the applications, being responsible for the authentication and authorization based on public keys cryptosystem.

To avoid communication among peer based in flooding technique, that waste bandwidth and is not deterministic for searches, it is assumed that peers apply a distributed repository to share common information among them regarding security, identification, localization of content etc. Peers regularly publish their available resources to the P2P network in shared repository. The publication describes document content (through keywords) and identifies document location for downloading, for example.

Initially it is assumed that all peers have a public key to iself identification in the P2P level application – this identification is persistent and independent of the peer id that is utilized in the network level for routing, etc., meaning that nothing changes in the P2P network level.

The peer must also have a private key correspondent to the public key identification to be able to make digital signatures. Observing that the public key may only be utilized as a unique identifier – for identifying the peer in the Internet and preserves anonymity – because knowing the public key does not imply having access to the peer (identified by peer id). The peer id changes constantly (normally on such initialization of the peer), but in such case the peer publishes in the distributed index the new mapping from the peer id to the public key. The anonymity is preserved because a public key does not necessarily identify a principal (peer) in the real world.

Every time a peer intends to share a document it signs the content, that is stored in local peer repository, and publish the keywords describing the content in the distributed repository (shared index). Thereby, the peer that queries the P2P network knows whether the peer that is doing the publishing is a trustworthy servant. The authenticity of publication (digital signature verification) can prove the clue of content authenticity, but an evaluation done based on the reputation of the key that is publishing, for example, can be more accurate on rating; if one peer begins to behave in an unexpected way it is a clue that it can be compromised.

As in the proposal everything must be signed (publication, content and certificates), easing the difficulty of the non-repudiation mechanism. Signed publication avoids false denial of source (non-repudiation of origin) and the peer that supplies content (server) record (log) the peer (key) that downloads content (non-repudiation of destination). If there is the need of arbitration due to non-repudiation, the anonymity of the reader may be disclosed in the server where the content was downloaded. In other cases, all types of anonymity related in [1] are preserved.

Access control to P2P content is not very common in conventional media sharing (music, films etc.); however, in professional applications that use P2P *middleware*/infrastructure as support for distributed environments, scalability, etc., it makes a sense. In this case, authorization certificates need to be applied. A peer (issuer) who wishes to control access to content, publish it enciphered into the public key of the destination (subject). Additionally, if content update is allowed the issuer needs to publish the authorization certificate – designated to the delegation rights directly to the public key (subject) that will have permission to modify the P2P content. The subject key updating the content sign the modified content and the certificate of authorization that will compose the authorization chain [23], attach it to the last certificate of the authorization chain and publish it on the shared index directory.

The P2P infrastructure layer offers resources for storage and transportation of P2P objects, abstracting physical network (infrastructure of communication) to the higher layers. Furthermore, P2P offers secure communications channels using cryptography, further than basic resources for the operation of a peer in a P2P network [26].

Prevention against denied of service attacks or other attacks on the network level such as exploits and other types of *malwares* (*malicious softwares*) is not the objective of this work.

Peers may choose to download content only from other peers who already know the public key (through peer reputation), since the probability of getting false content from an authentic publishing of the peer is very low. Therefore, the credibility of the public key that publishes is based on its positive reputation with the client peer. The positive reputation of a key is built based on supplying authentic content, because a peer may provide authentic publishing (with verified digital signature) but with false content. The evaluation of content authenticity can only be done by a human [3].

Evidently, a peer client may associate a good degree of credibility for a public key without having a positive reputation of it, whether the key is recommended by a peer that already has a good reputation with that client.

The publications shared by all in the repository can also be applied to keep a chronological authenticity of publication [3], preventing an already published content from being illegally republished as new by a malicious peer.

The content authored by peer is identified by the format: keyAuthor@documentName (document identification). When a peer has a copy of content published by another peer, it will announce a new content identification by the format: keyServer@*keyAuthor@documentName* (replicated document identification).

Certificates apply the URI in the delegation field to identify the document (object) of authorization. As a URI does not make sense in a P2P network it adopted the usage of public key concatenated to document name, aforementioned as document identification, in replacement to the classic use of a URI in that type of certificate.

The proposed reputation scheme is based on qualification of both author and peers replicating content through a voting system (Figure 4). A peer replicating content is a node that stores original copies of content produced by a peer author of content. It is easy for a peer client to differentiate copies (replication) from original content of document through the document identification

When a peer requests content (document) for downloading from a peer server, the server sends to the client a qualification request. That request must be signed by the client and returned to server in order to obtain the document; the server publishes the qualification request on the shared repository and provides the document to the client. After downloading the document the client evaluates the content and attributes a grade for author and server of the content through voting that is stored on the shared repository. The grade can be neutral, positive or negative, ranging from neutral to highest positive/negative value. On the shared repository a qualification request is answered by the respective voting expressing the grade associated by the client to author and server of content; the voted answer except the respective qualification request.

When a client peer attributes a positive grade to author/server of content, it is must share that content on the shared index, becoming itself a replicating peer server; otherwise the replication is not recommended in order to avoid junk content replication.

Based on the reputation scheme a P2P score service (Figure 2) is being proposed. The service frequently (in regular, configurable, periods as an hour, day, week etc.) queries the shared repository to collect qualification data and produce (statistics) reports about qualification requests pending and positive, neutral or negative qualification that imply in reputation rating (per peer). Additionally, the data collected are evaluated to identify free rider behavior (i.e. a peer that issues more negative or neutral qualification than positive to avoid sharing resources with the network).

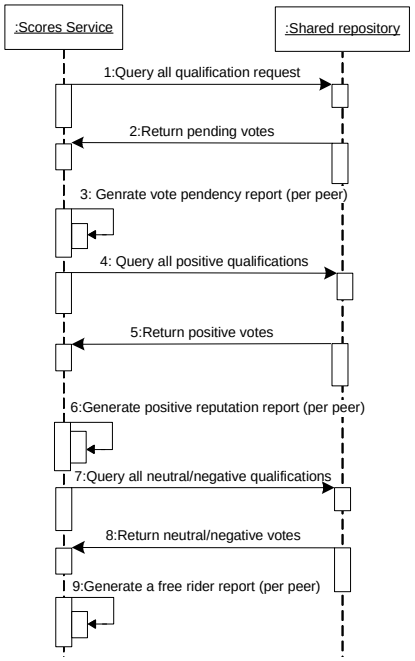

Figure 2. P2P Scores Generator Service

All the scores produced about a peer can be used as attribute to be applied on usage control as a P2P compensation for good reputation. That is, usage control is independent of access control in the sense of specifying usage policies that allow a client to download content only whether their scores are suitable to the requirements of usage policies. A peer server can or cannot adopt usage control. In fact, usage control aims at stimulating the sharing of content (documents), fighting against free rider behavior.

A. Dynamic of the Model

The main purpose of the proposal is to provide support to publishing and verification of publications authenticity on the shared index. Additionally, the credibility of servant in providing authentic content can be determined based on the public key reputation scheme, as well as, whether a modification in content kept its authenticity through the certificate authorization chain.

In a nutshell publications are made on a repository (that plays an index role), through publishing, using keywords to describe the document's content (Figure 3). Peers that query the repository (searching by content associated to publishing)

may check the authenticity of one publication verifying its digital signature. After downloading the content from the peer server it is possible evaluate its authenticity and answer the qualification request on the shared repository.

Figure 3. Overview of Exchanges Among the Entity of the Model

If a peer wishes to know who the key performing publishing belongs to, they may search the name certificate associated to it in the index repository. If the certificate exists in the index, the peer can identify the author of the publication; otherwise the publication is anonymous, which does not mean that the anonymous publication cannot be authentic. A key can make authentic publications that will also have an authenticated digital signature; however, the author prefers not to be identified.

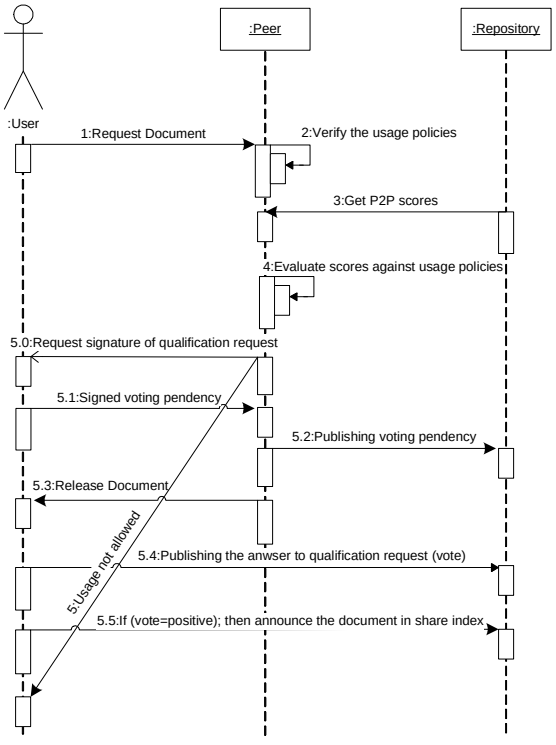

Figure 4. Usage Control and P2P Reputation System

Figure 3 shows the P2P network entity which represents an abstract network overlay, in this case, the access of document searching is actually being done in Peer B, but the P2P network abstracts this type of knowledge from the nodes downloading document/content (peer A).

The access control uses cryptography to constrain access to P2P content, given by default all peers of the network have reading access to the content available in it. That means, when content can only be modified (written) by a principal, the content must be encrypted in the subject's public key, stored locally and the authorization chain is published with the same identification of document identification in the shared index (that is additionally made to step 3 of Figure 3). After updating the document the principal (subject) can publishe the keywords and the authorization chain in order to provide authenticity to the modifications and preserve the author of document. In that case, the authenticity of the authorization chain is an additional verification accomplished in step 8 of Figure 3.

Before supplying a document to client download, the server can get, from scores service, a report about peer client "relationship" with the network. The scores can be applied to confront usage policies against peer scores and decide if the usage will be allowed or not. If so (step 5.0, figure 4) the document is supplied and the scheme to update reputation is performed. Otherwise, the usage will not be allowed (step 5, figure 4) and the process is finished.

One can notice that a positive qualification of author/server of content imposes to downloaded client sharing content – to avoid free rider behavior (step 5.5, figure 4).

B. Implementation Issues

The prototype architecture is composed by various technologies which jointly implement the proposed model. P2P infrastructure of Jxta was used to achieve platform and network environment interdependence, as well as to provide a transportation means for P2P objects. The SPKI/SDSI was used to serve as a security and distributed infrastructure to provide access control, certificates, public keys and authenticity. The certificates repository and certificates search engine, equivalent to a directory services, with addition of being distributed and scalable, is obtained from the using the DHT based on Bamboo implementation [19]. DHT is also applied to the shared index/repository, according to figure 5.

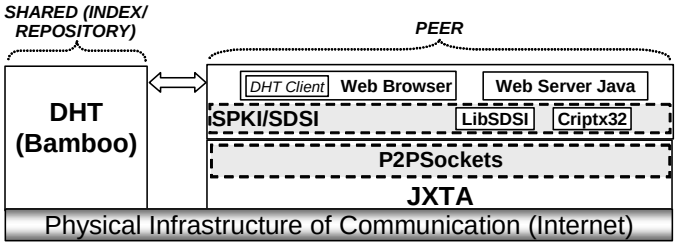

Figure 5. Prototype Architecture

In the application layer a browser written in Java is developed with an embedded DHT client, it represents the P2P client. The browser was developed from Java classes and has a html editor incorporated to facilitate the execution of the case

study. The Apache web server Jetty was used as the P2P server in the application layer; all the documents/contents stored locally are available for download through the web server.

The browser accesses SDSI/SPKI facilities (through plug-ins) to check signature authenticity and authorization. That is, the SDSI/SPKI offers facilities for digital signature, key generation, and name/authorization certificates handling based on SDSI and Criptx32 libraries. Furthermore, SDSI/SPKI offers facilities for access control (enforcement), such as verification of certificate authorization chain sequence. When updating a content, the browser allows editing and document signature, besides attaching the authorization certificate to the chain that will publish it on DHT (shared index).

The p2psockets offer software that adapts the classic Java socket implementation in order to use Jxta infrastructure, i.e. programming with Java sockets abstract the existence of Jxta by the usage of P2PSockets. On the prototype, Jxta was applied to P2P content transportation and for abstracting the P2P environment.

C. *Scenario*

Considering a news agency where all reports are made available online using the Internet and that avoids the costs of high availability systems, the unique point of failures using a central server and dependency of a *web designer* the agency chose apply to P2P network. P2P allows quick availability of news in a competitive area, since being the first journalist to publish news affects career success. The agency staff (journalists and editors) makes available reports on their own computers. A journalist can be in the most remote place when producing news, however, making the news available, allows immediate reading without sending it to news center to be edited and designed, and afterwards published on web page.

If an approval or revision by the editor in chief or something similar is required, the news (document) can be enciphered into the public key of editor, stored locally and the keywords and SDSI/SPKI authorization chain published on DHT (share index). The editor can get the document; review it, store locally and publish the same keywords and authorization chain in the index. One searching shared index finds the document (news) and authorization chain, which keep the author of the news – the journalist and not the editor in chief (due the first certificate of the authorization chain).

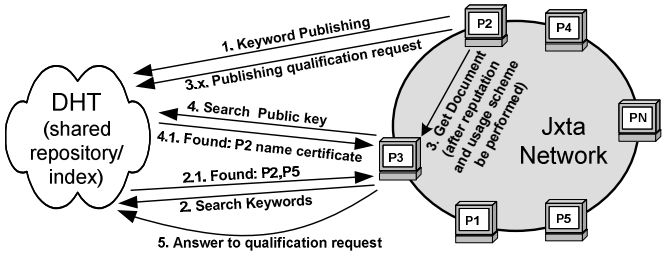

Figure 6. News Agency Scenario

Initially, it should be considered in the context above that all journalists that wished to publish their news should obligatorily have a pair of keys and preferably a certificate of SPKI/SDSI names published in the shared index – the certificate facilitates the identification of the journalist.

From the implemented browser in the prototype it is possible to edit html pages (the editor is simple, however it allows evaluation of the prototype). After editing the news in the HTML editor, a plug-in is triggered to digitally sign the report. After signing the news the journalist store it into web server share directory, creating and signing the keywords and document identification and publish them on the shared index (Figure 6, step 1).

At a given time a P3 peer, for example, searches for some keywords and finds various publications (Figure 6, step 2), among them it is the one previously published by P2. As P3 needs to choose among various publications returned (from index search), let's assume that P3 has already obtained content from P2 previously; therefore P2 is on the P3 credibility reputation list. Thereby, P3 requests the news from the web server running in P2 through the Jxta network (Figure 6, step 3). In fact, in order to get the document the steps presented in figure 4 must be performed.

The web server records the public key of the P2 peer to avoid non-repudiation. The anonymity [27] of P3 is not being violated because if P3 has not published a names certificate, there will not be name linking the peer to its identification (public key). After downloading the news via http, P3 records its source (for non-repudiation) and verifies the digital signature of the document. It is important that P3 logs the public key of document source because whether the document is replicated, the peer server of replicating document could be lost. The recording of document source is applied to update the credibility reputation list and for non-repudiation purposes.

After reading and evaluating the download news, P3 may wish to know to whom the public key that signed the document belongs to; in this case P3 must search the shared index again in order to retrieve the names certificate correspondent to the P2 public key (Figure 06, step 4).

If the document is authentic, P3 attributes a positive grade to the (author) public key and for the peer replicating server and issue the answer to qualification request on the shared repository. The replicating server that shares non authentic documents will have its credibility reduced by attribution of negative qualification.

This context was implemented using the technology cited in the implementation aspects section (5.2), however these cases are not limited to the mentioned application but cover any scenario where the proposal is feasible.

V. RELATED WORKS

In the technical literature many works are focused on proposed techniques to assure security proprieties to the content distributed in the P2P networks.

The authenticity of content (document) can be defined in multiple ways according [3], by: the oldest document (it is assumed that the first posted is authentic), expert-based (the content is evaluated by an expert that issues their opinion about the authenticity of content) or voting-based (this approach differentiates itself from expert-based since a set of

experts vote to decide the authenticity of a content.). Additionally, according [3] the authenticity of content can be evaluated by reputation based mechanisms, that aim to collect and share opinions developed between peers about their trustworthiness when sharing resources. To track peer reputations in a centralized P2P network like Napster is not difficult because in a central server the search for content is facilitated by the server. But in a decentralized P2P network it can be a problem to accurately track reputation.

In [35] there are two approaches present for reputation; debit-credit and credit-only. Debit-credit mechanism credits peer reputation scores for serving content and debits for downloading. The credit-only mechanism credits peer reputation scores for serving content but offer no debits; this proposed reputations system focuses on unstructured, decentralized P2P networks like Gnutella.

To assure content authenticity in the Poblano project for Jxta platform [36], the name certificate issued by a CA (Certificate Authority) or auto-signed certificates to make digital signatures based on public keys obtained from that certificate are applied. Poblano also can manage the certificate based on a trustable network similar to the concept of Web of Trust of PGP (Pretty Good Privacy) making a chain of recommendation in this way [37].

EigenTrust project [33] uses the concept of transitive trust for reputation of peers and contents. Meaning that a peer considers the opinions of peers it trusts more, for example, if a peer A trusts any peer X, it would also trust the peers trusted by X. For each peer A calculates the local trust value for all peers that have provided it with authentic or fake downloads based on satisfaction feelings in the transactions carried out among them.

In Credence [32] the focus is on the content reputation. Credence counteracts pollution in P2P networks by allowing honest peers to assess the authenticity of online content through secure tabulation and management of endorsements from other peers. Also Credence employs a novel voter correlation scheme to weight peer opinions, this approach give rise to favorable incentives and system dynamics.

After getting a file, a user of a Credence enabled system is given the chance to submit a single vote. Either a positive (thumbs-up) vote for content matching its description or a negative (thumbs-down) vote for pollution. Votes are cryptographically signed, when a user asks for votes on a file, positive and negative opinions are counted respectively.

Xrep project is a Gnutella protocol extension [31], unlike previously described reputation systems, not only a peer reputation system, but also content reputation, for this reason it is a fully distributed reputation system. When one peer request a file, all peers who have the matched keywords, respond to the query including a file's digest. To select the proper peer for downloading, a new query is made for other peers, asking for the reputation of the candidate server peers and their files. These other peers respond with their IP address and their opinion about server peers and their respective files. The peer that requested all opinions judges the reputation and then after downloading the file from the selected peer, the file is checked against its digest to ensure the integrity and then the peer will update its repositories with its opinion on the downloaded resource and its providers (server peers).

A simple way of identifying a peer is by IP address. However this method is severely limited because they are vulnerable to IP-Spoofing as well as peers frequently having IP addresses dynamically associated by their ISPs. Instead, a more reliable method may be to use self-signed certificates. In the Identity Crisis model, that assumes all peer use the same identity for their lifetimes, this model uses self-signed certificates, allowing well-behaved nodes to build trust between each other during a series of disconnections and reconnections from different IP addresses [7].

The proposal presented in [34], based on the prisoner's dilemma, makes an analysis of the social cost in allowing nodes to freely change identities. Thus, the project creates a mechanism based on a centralized trusted intermediary and assures that a user is assigned only one system identifier; at the same time keeping the user anonymous because the trusted intermediary does not know which identifier was assigned to which node.

According to [8], anonymity can be viewed in many aspects, making difficult for peers to find out who created a file, who stores a file, who accesses a file and which documents are stored on a peer. The authors, also, consider that anonymity is seen as an advantage in P2P, since it can open doors to various security problems that can alarm users. It is perfectly reasonable to trust a single centralized service, but obviously unwise to trust any multitude of anonymous resource-providers in the whole P2P network, a malicious peer can easily deceive other peers, and hackers as well as worm viruses can use spoofed identity to damage the whole P2P system. The author suggest the adoption of asymmetrical keys which do not disclose anonymity, this way node identification is made using its own key or hash. Additionally, the authenticity of content can be reached by digital signature.

In the works of [28] the issues of authentication and anonymity are considered where an authentication protocol of P2P systems was created. The protocol is based on Merkle's Puzzles allowing a secure communication between two parties and Zero-Knowledge Proofs which is a protocol that allows the node to have ownership or knowledge of a "secret" to convince the "verifier" that does the authentication, however without revealing the whole secret. In this project Anonymity is obtained with a packet-preemptive proxy service model technique that was proposed based on Gnutella protocol, the packets used are Query, QueryHit, and Push.

Access control deals with restricting access to resources to peers that have the right to access those resources. In [11] an approach for providing the strong and efficient access control mechanism based on RBAC, to P2P systems is introduced. This model supports autonomous decisions and centralized controls, in other words it can work with both P2P models, brokered as well a purist, the architecture is designed and developed as a middleware platform and works like a broker between peers, providing a controlled P2P environment.

In a nutshell, the proposals found in technical literature bases the P2P security on client-server secure architecture adapted to P2P limitation. The usage of key to identification and digital signature to authenticity of content are not new, since it is easily adapted from client-server architecture. However anonymity, non-repudiation, usage control, distributed compensation scheme based on scores and access control based on public key to P2P were not mentioned in the literature, as proposed by us. Additionally, in our proposal we present a scheme to give creditability to auto-signed certificates and applied a PKI (SDSI/SKPI) that is more suitable to P2P features, mainly, due its ability to manage authorization without applying TTP.

VI. Proposal Considerations

As aforementioned, false or corrupted content (polluted) grow exponentially and already represent more than 50% of available P2P network content. Polluted content traffic diminishes the amount of bandwidth available for healthy network use.

In the proposal everything is digitally signed, moreover with a scheme of credible public keys, it is expected that only authenticated and signed content will be shared, because public keys that do not have credibility will not be accessed; therefore they will have no benefit in sharing polluted content. This tends to diminish the effect of free rides [28] that only consume from the network and do not make anything available for the network. Nowadays the P2P system attempts to minimize that behavior (free rider), requiring compensation of the peer, i.e., in order for a peer to obtain content from the P2P network, that peer needs to share similar amounts of content with it. In that case, many bad peers make big files available with fake content to gain a similar amount of content from the P2P network. By using the proposal the content from a bad peer will not be downloaded, because a peer that shows that behavior will not appear in the positive reputation of any other peer in the P2P network.

The usage control acts as an reliable P2P import compensation mechanism due the absence of a centralized point to control free rider behavior.

Identification through public keys guarantees the persistent identification of peers/principal, however preserving the anonymity of those peers that do not wish to be identified.

Access control is done by peer target of delegation through an authorization chain, assuring that the ciphered content arrives to the addressed peer without allowing intermediate nodes to read the content; a bad node could copy the content and publish it as new content, authored by it. Furthermore, it preserves the identification of the author of the document through the certificate authorization chain.

Access control in the proposal is not based in ACLs and trust with a TTP (typical of client-server architecture) and therefore it is more appropriate for the dynamic features of P2P network.

The state of confidentiality is also accomplished with this proposal because content may be ciphered and only the private key owner will be able to decipher the document, thus, even using an original feature of the P2P system that is file sharing, the security mechanism guarantees that only authorized peers have access to the content.

Besides confidentiality there is another important property, integrity which is obtained by the use of associated hash mechanisms to digital signature, it is possible to identify whether or not a document was modified.

The scenario shown above has additional advantages to those cited in that section, for example, common sites publish a news digest and create a link to the original content. Sites that regularly suffer updates on content and a previously linked page can change their URL, therefore the link becomes lost. A more prevented news digest publisher may do a local copy of the news, in this case, the copy could not be authenticated and the news author may be unfairly missed. With the proposed project, anyone will be able to keep an authentic copy of the content locally and more importantly, an authentic authored copy.

The adoption of a non-repudiation mechanism avoids a journalist falsely denying published news, for example.

The use of digital signatures, storing content locally and publishing keywords is a procedure that minimizes accidental sharing, since it is not only a copy of content in a directory (as is the common procedure for most current P2P clients), but it is a process to be performed.

It is important to point out the shared index is not an index server, it is a distributed and fault tolerance repository, not figuring as a possible central point of failures or vulnerabilities.

The credibility scheme can be seen playing the role of the CA on PKI X.509, for example, because a positive history of a peer endorses its public key.

VII. Conclusion

Public key as a persistent peer identifier allowed effective control against polluted content dissemination, also reducing negative effects that free rides nodes bring to the network. The sharing of authentic content also prevents the distribution of malware whether the sharing is of executable codes, for example.

Peer reputation provides the required credibility to the public keys in order to give trustworthiness to auto-signed certificates.

The proposal presented an alternative for P2P access control based on public keys and a scheme for the verification of P2P content authenticity. With the use of certificate authorization chain it was possible to grant rights to modify replicated content in P2P networks without losing authenticity and preserving the author of content.

The prototype allowed evaluation of URI replacement efficiency – in identifying objects protected by SPKI – by an identification mechanism of independent objects from the server path, based on the linking of the public key of the peer with the document name.

The prototype showed that the scenario with the P2P based news agency is advantageous in comparison to the conventional one. The main advantages are immediate content

availability without the need of intermediation of web designers, and principally by authentic content availability even outside of the agency's site.

The section proposal considerations brought other important issues that are covered by the proposal.

REFERENCES

[1] *Dingledine, R.; Freedman, M. J.; Molnar, D.. Peer-To-Peer: Harnessing the Power of Disruptive Technologies, chapter 12: Free Haven. [online] available in: http://www.freehaven.net/doc/oreilly/freehaven-ch12.html, last access: 05/2006.*

[2] *Kumar, R., Yao, D., Bagchi, A., Ross, K., Rubenstein, D. Fluid Modeling of Pollution Proliferation in P2P Networks, In: Proceedings of ACM Sigmetrics, 2006.*

[3] *Daswani, N.; Garcia-Molina, H.; Yang B.. Open problems in data-sharing peer-to-peer systems. In: Proceedings of 9th International Conference on Database Theory, volume 2572 of LNCS, pages 1-15. Springer, 2003.*

[4] *Wölfl, T. Public-Key-Infrastructure Based on a Peer-to-Peer Network. In: proceedings of 38th Hawaii International Conference on System Sciences, IEEE Computer Society.*

[5] *Berket, K.; Essiari A.; Muratas, A..PKI-Based Security for Peer-to-Peer Information Sharing, In: Proceedings of P2P' 2004.*

[6] *Zhang, X,; Chen, S.; Sandhu, R.. Enhancing Data Authenticity and Integrity in P2P Systems, In: IEEE Internet Computing, nov-dec, 2005.*

[7] *Marti, S.; Garcia-Molina, H.. Identity crisis: Anonymity vs. reputation in p2p systems. In: IEEE 3rd International Conference on Peer-to-Peer Computing, IEEE Computer Society, 2003.*

[8] *Tang, L.. Identifying Resource Authenticity in P2P Networks, In: Proc. of the 2nd International Conference of Applied Cryptography and Network Security, 2004.*

[9] *Crispo, B.; Sivasubramanian, S.; Mazzoleni, P.; Bertino, E.. P-Hera: Scalable fine-grained access control for P2P infrastructures, In: Proceedings of 11th International Conference on Parallel and Distributed Systems (ICPADS), 2005.*

[10] *Zhang, Y.; Li, X. Huai, J.; Liu, Y.. Access Control in Peer-to-Peer Collaborative Systems, In: 25th International Conference on Distributed Computing Systems Workshops, IEEE Computer Society, pg 835-840, 2005.*

[11] *Park, S. J.; Hwang, J.. Role-based access control for collaborative enterprise in peer-to-peer computing environments. In: Proceedings of the eighth ACM symposium on Access control models and technologies. pg. 93 – 99, 2003.*

[12] *C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, SPKI Certificate Theory, RFC 2693, September 1999, 1999.*

[13] *Aura, T.. Fast Access Control Decisions from Delegation Certificate Databases. In: proceedings of 3th Australasian Conference on Information Security and Privacy, Berlin, 1998.*

[14] *Santin, A. O. ; Fraga, J. S. ; Maziero, C.. Extending the SDSI / SPKI model through federation webs. In: Proceedings of Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security. LNCS 2828, Springer-Verlag, p. 132-145, 2003.*

[15] *Junginger, M.; Lee, Y.. A Self-Organizing Publish/Subscribe Middleware for Dynamic Peer-to-Peer Networks. In: IEEE Network, Vol. 18, n° 1, pg. 38-43, 2004.*

[16] *Lawton, G.. Is Peer-to-Peer Secure Enough for Corporate Use? In: IEEE Computer, January 2004.*

[17] Resnick, P.;Zeckhauser, R.; Friedman, E.; Kuwabara, K. Reputation systems. Communications of the ACM, pg. 45-48. 2000.

[18] *Rüdiger Schollmeier. A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Application, In: First International Conference on Peer-to-Peer Computing (P2P`01), Linköpings universitet, Sweden, 2001.*

[19] *S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, OpenDHT: a public DHT service and its uses, In: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 73-84, 2005.*

[20] *Traversat B., Arora, A., Abdelaziz, M., Duigou, M., Haywood, C., Hugly, J., Pouyoul, E., Yeager, B. JXTA 2.0 Super-Peer Virtual Network, [online] available in: http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf last access in: 07/06.*

[21] *The Internet Engineering Task Force. SPKI Requirements. [online] http://www.ietf.org/rfc/rfc2692.txt last access in: 04/06.*

[22] *Clarke, D. E. SPKI/SDSI HTTP Server Certificate Chain Discovery in SPKI/SDSI. Department of Eletrical Engeneering and Computer Science of MIT, master dissertation, 2001.*

[23] *The Internet Engineering Task Force. SPKI Certificate Theory. [online] http://www.ietf.org/rfc/rfc2693.txt. 1999.*

[24] *Lampson, B. e Rivest, R. L. A simple Distributed Security Infrastructure. [online] http://theory.lcs.mit.edu/~cis/sdsi.html.1996.*

[25] *Elien, J. E. (1998). Certicate discovery using SPKI/SDSI 2.0 certicates. Department of Eletrical Engeneering and Computer Science of MIT, master dissertation.*

[26] *C. Schmidt and M. Parashar. Enabling Flexible Queries with Guarantees in P2P Systems. In: IEEE Internet Computing, 3(8):19--26, 2004.*

[27] *Kim, B. R., Kim, K. C., and Kim, Y. S. 2005. Securing anonymity in P2P network. In: Proceedings of the Joint Conference on Smart Objects and Ambient intelligence: innovative Context-Aware Services: Usages and Technologies. sOc-EUSAI '05, vol. 121. ACM Press, New York, NY, 231-234, 2005.*

[28] *Wierzbicki, A.; Zwierko, A.; Kotulski, Z. Authentication with controlled anonymity in P2P systems. In: Proceeding of Parallel and Distributed Computing, Applications and Technologies. Sixth International Conference on Volume, Issue , 05-08 Page(s): 871 – 875, 2005.*

[29] *Tang, Y., Wang. H., Dou, W. Trust based incentive in P2P network. In: E-Commerce Technology for Dynamic E-Business. IEEE International Conference on September, 2004.*

[30] Damiani E., De Capitani di Vemercati S., Paraboschi, S., Samarati, P. and Violante, F. A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. Proceedings of the 9th conference on computer and communications security, 207-216, 2002.

[31] K. Walsh and E. Sirer. Experience with an object reputation system for peer-to-peer filesharing. In Networked Systems Design and Implementation, NSDI, 2006.

[32] Kamvar, S., Schlosser, M., and Garcia-Molina, H. The EigenTrust algorithm for reputation management in P2P networks. Proceedings of the Twelfth International World Wide Web Conference, 2003.

[33] Friedman, E. and P. Resnick (2001). The Social Cost of Cheap Pseudonyms. Journal of Economics and Management Strategy 10(2): 173-199.

[34] Gupta, M., P. Judge, and M. Ammar. A Reputation System for Peer-to-Peer Networks. in ACM NOSSDAV. California, June 2003.

[35] Chen R., Yeager W.: Poblano: A distributed trust model for peer-to-peer networks. Technical report, Sun Microsystems. http://www.jxta.org/docs/trust.pdf, Maio 2005, 15:00.

[36] Walsh, Kevin, Sirer, Emin Gün: Experience With A Distributed Object Reputation System for Peer-to-Peer Filesharing. To appear in Proceedings of the Symposium on Networked System Design and Implementation (NSDI), San Jose, California, May 2006.