Implementation of PKIX-P2P with trust-based reputation using the JXTA platform

Francisco Sales de Lima Filho
IFRN
Natal - Brasil
sales.filho@ifrn.edu.br

Sergio Vianna Fialho
UFRN
Natal - Brasil
fialho@pop-rn.rnp.br

Abstract— **Nowadays, implement security in P2P systems is a great challenge. The possibility of building an infrastructure of public keys (PKI) operating in P2P environments represents a great advance. In this context, this paper presents an implementation of a P2P application with trust-based reputation using the JXTA platform.**

Keywords-component: P2P, PKI, PKIX, Trust-based reputation, Information Security, Digital certification, JXTA.

I. INTRODUCTION

The P2P (peer-to-peer) evolved greatly in recent years. Systems as resource sharing, messaging and collaborative works are typical examples of P2P applications. Along with the growth of P2P applications, are emerging threats and vulnerabilities to this new tools. Thus, seeking to maintain the information security principles, we must develop mechanisms aimed at ensuring the safety requirements for this new generation of applications.

The solutions of infrastructure of public keys used in common applications are based on client/server architecture, not adhering fully to the dynamic environment built by the new P2P systems.

In this context, this paper proposes the use of a PKIX-P2P (P2P public key Infrastructure based on X.509 certificates), using a parameterized reputation mechanism to measure trust in this system.

This paper was written to provide concepts about communication in P2P environments, public key infrastructure and basic operation, as may be seen in sections II e III define the operation of PKIX-P2P and the reputation protocol in the section IV and displaying the results in section V.

II. PEER-TO-PEER COMUNICATION ENVIRONMENT

In recent years, P2P networks have aroused the interest of many researchers in the fields of computing, computer networks, distributed systems, database, among other. This interest has contributed greatly to the development and improvement of P2P systems. Currently, several criteria are used to characterize and define a P2P network topology as semi-centralized or decentralized. Some of these criteria are specified in [1]:

- The network should be composed of leaf nodes.
- The nodes can join or leave the network frequently.
- The node´s addresses can change, since it leaves the network.
- The nodes must have partial or full autonomy in relation to a centralized server.
- The network must be highly scalable.
- The nodes can communicate directly with each other without the intervention of a central server.

A system with these characteristics can be called P2P, even though some of the network control functions are located in a special node, which functions as central server (point of failure).

P2P networks are also called overlay networks, for operating on an infrastructure of existing network, forming interconnections between the various peers. In the case of this work, the basic infrastructure is provided by TCP/IP architecture.

The JXTA platform defines and implements an overlay network, designed as an open solution to provide communication between nodes in P2P networks. Currently, the JXTA project it´s in the version 2.5 and provides a complete platform for developing P2P applications, with a clear set of protocols to perform the core operations:[2]:

- Peer discovery.
- Group-based organization.
- Advertisements and service discovery.
- Direct communication between peers.
- Monitoring of peers.

The Figure 1. presents an overview of the JXTA platform and the mapping of virtual network nodes to existing physical network.

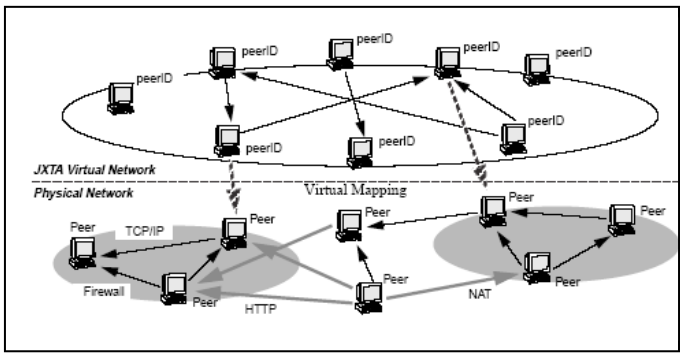

Figure 1. Overview of the JXTA virtual network.

JXTA peers use pipes to send messages to one another. Pipes are an asynchronous, message transfer mechanism used for communication and data transfer. Pipes are virtual communication channels and may connect peers that do not have a direct physical link, resulting in a logical connection [2].

Use the resources offered by P2P technology has been the subject of many studies, either in developing new specific P2P applications or adapting traditional client/server applications. In the information security area, specifically in the public key infrastructure, the work of [3] has an important contribution.

Thus, we conclude this section saying that P2P networks provide a rich environment in the development of security applications, particularly in meeting the requirements of availability and fault tolerance.

III. BASIC PRINCIPLES OF KEY INFRASTRUCTURE

In the era of digital certification, the mechanism for public key infrastructure serves as virtual office, doing management of digital certificates through the operations of creation, validation and revocation[4]. The certificates are standardized data structures that store important information about the issuer and the user, such as digital signature, expiration date and key (s) of encryption. The main objective of this infrastructure is to provide mechanisms to verify the certificates as to the authenticity and validity. Typically, a user Bob presents his credentials (public key) to Alice, who must decide if trust or not on credentials based on the following questions:

a) Is the issuer´s public key authentic?

b) Is the certificate still valid (not revoked and not expired)?

c) Does Alice trust the issuer of the certificate?

The issuer´s authenticity, question "a", can be made searching on certification chain until the root node, anchor of the infrastructure that has a well-known public key. The validity (question "b") is checked, usually by consulting the list of revoked certificates, provided by the infrastructure, and the dates contained in the certificate itself. Finally, the decision to trust the issuer (question "c") emanates from Alice decisively. The trust is established, if all the checks are positive responses.

In centralized environments, based on client/server architecture, the information security principles, as availability, can be compromised and the checks fail, causing the failure of the system.

Deploy an infrastructure of decentralized public keys, where the certificate chain is flexible and fault-tolerant, indicating that this system fits into the P2P paradigm.

IV. KEY INFRASTRUCTURE IN DESCENTRALIZED ENVIRONMENT (PKIX-P2P)

The infrastructure of traditional public keys should be structured to be consistent with the types of individuals who should administer it, as mentioned in [4]. However, the concept of centralization does not apply directly to P2P environments, needing to adapt to the dynamics of this kind of network. In this context, P2P-PKIX (Public Key Infrastructure X.509 - Peer-to-peer) uses some of the elements defined in the PKI traditional (Public Key Infrastructure), like the concept of entities and certificate templates, adapting them to operate in P2P environments.

The JXTA platform [5], as a P2P structure for general use, was employed in the development of P2P-PKIX is described in this work, because of their degree of maturity, documentation, portability and interoperability of the Java programming language.

As in traditional PKI [4]., management protocols to support online interactions between the entities of the PKIX-P2P are needed, for example, to accomplish the exchange of certificates between final entities (users) and the Certification Authorities (CA) P2P.

The PKIX-P2P model adopted in this work is based on the proposal of [3], where each node of the P2P network is potentially a Root Certification Authority and can issue and validate certificates, differing, however, when define specialized nodes and more reliable for this function, while common nodes represent the P2P end entities, as shown in Figure 2.

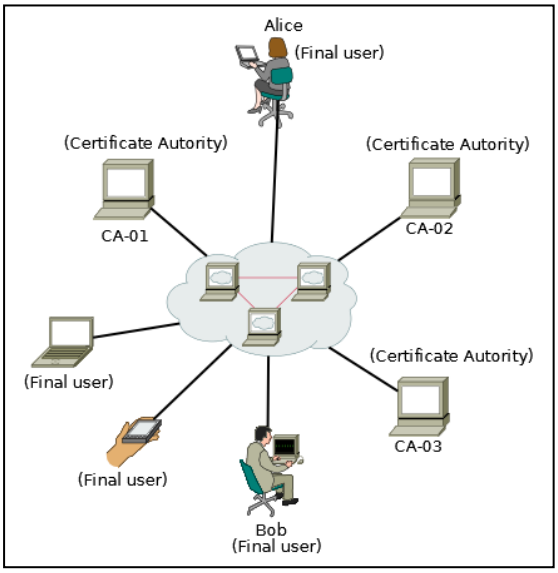

Figure 2. Overview of the PKIX-P2P.

The nodes that perform the PKIX-P2P JXTA may be Edge peer or Super peer, depending on the desired network topology.

A. *Operation Mode*

The beginning of the operation of the network is performed through the Original Certification Authority - OCA, which self-signed your own root certificate and publishes this feature through the P2P network, while carrying out searches for the discovery of other CA available.

New CA, in order to become members of the PKIX-P2P system, gets their root certificates already signed by the OCA or self-sign their certificates and announce its operation in the network. Communication between the CA is accomplished through a protocol that was developed for this purpose. Beyond messaging, the communication protocol is used to calculate how reliable is a CA, based on the reputation that it has on network. The reliability of the new CA will be build from the views of other participants of the system.

B. *Reputation and replication protocol*

Mechanism used in communication between the CA, the protocol also measures the reputation level that a particular CA has in the system, based on previously established parameters. The calculation of the reputation is established considering the following parameters:

- Administrative parameter (configured by the CA administrator).
- Time on system (calculated in proportion to the age of the system).
- Certificates issued.
- Availability index (Automatically calculated by the reputation protocol).

The reputation of a CA is represented by a positive real number between 0 and 1 where 0 is a lower reputation and 1 is the maximum. The calculation is done locally by each CA according to the eq. 1.

$$R_{CA} = \{[(MR_{CA} / T_{MR}) * W_{MR}] + [(CI_{CA} / T_{CI}) * W_{CI}] + [(AS_{CA} / T_{AS}) * W_{AS}] + [AD_{CA} * W_{AD}]\} / W_T \quad \text{(eq.1)}$$

Where:

R_{CA} – CA Reputation.

MR_{CA} – Reply Reputation messages.

T_{MR} – Total of reputation messages.

W_{MR} – Weight assigned to the parameter "received messages ".

CI_{CA} – Certificates issued.

T_{CI} – Total of issued certificates.

W_{CI} – Weight given to issued certificate.

AS_{CA} – Time on system.

T_{AS} - Age of system.

W_{AS} - Weight given to seniority parameter.

AD_{CA} – Administrative parameter.

W_{AD} - Weight given to Administrative parameter.

W_T – Total weights sum.

The replication process "piggybacks" on messages exchanged between the CA. The information may be are public or trusted by peers and, then, shared to all nodes participating of the PKIX-P2P. The Figure 3. and the following steps summarize the operation of the protocol:

1) Send discovery request.
2) Receive discovery response.
3) Send reputation request.
4) Receive response(s).
5) Information processing and calculation of reputation.

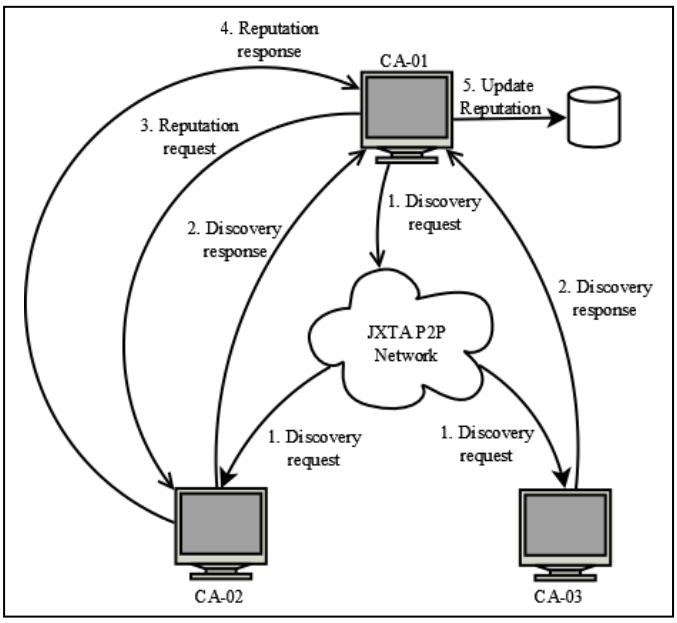

Figure 3. Overview of protocol reputation.

The reputation and replication protocol uses the same message structure, both for reputation´s request as well as to the response, changing only the *type* attribute, as can be seen in Figure 4.

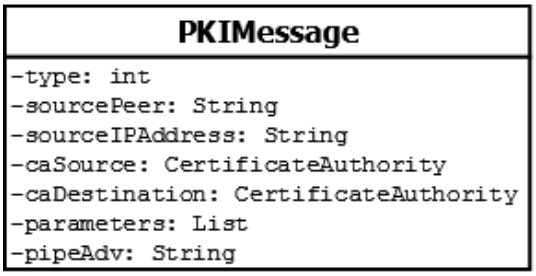

Figure 4. Message structure of protocol reputation.

When the *type* attribute is set to the value 1, it is a reputation request, when the value is 2 it is a reputation response . The *sourcePeer* attribute specifies the peer name on P2P network, while the attribute *sourceIPAddress* specifies the peer IP address that is sending the message on TCP/IP network. The attributes and *caSource caDestination*, type *CertificateAutority,* carry objects that identify the peers as CA on system. The *parameters* attribute is a generic container used by reputation protocol for carrying objects serializable, for the PKIX-P2P are transported *CertificateAutority* objects.

The final entities also uses the same message format that reputation protocol, changing only the semantics. In this case, if *caSource* field is *null*, then this is a simple reputation request made by final entity. If *caDestination* field is *null*, then the requestor accepts response from any CA, otherwise, just accept response by CA specified in this field.

C. *Final entity and reputation protocol*

Final entities make use of the reputation protocol in time to verify the trust in a certificate, analyzing the validity and authenticity of the information, and the reputation of the CA that issued the certificate. Then decide whether or not trust on the credentials presented. In Figure 5. , Alice must decide if trust or not in Bob´s certificate.

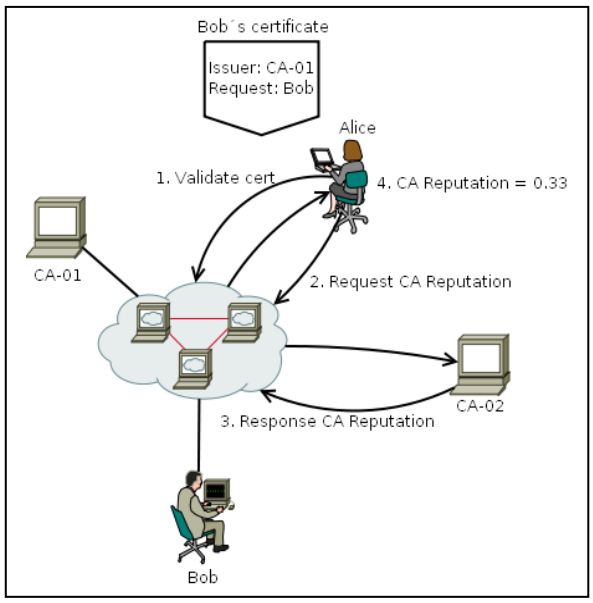

Figure 5. End entity using the reputation protocol.

In the above situation, considering that the public key of CA-01 is authentic and Bob´s certificate is valid (not expired or revoked), Alice can request information about the reputation of the CA-01 on the PKIX-P2P before trust in Bob´s certificate.

V. RESULTS

As the P2P communication model presented above, this work implements a public key infrastructure based on X.509 certificates and a communication protocol used to exchange messages between CA. This protocol define a reliable parameter called reputation. The application was developed using the Java language, JXTA API and modeled according Figure 6.

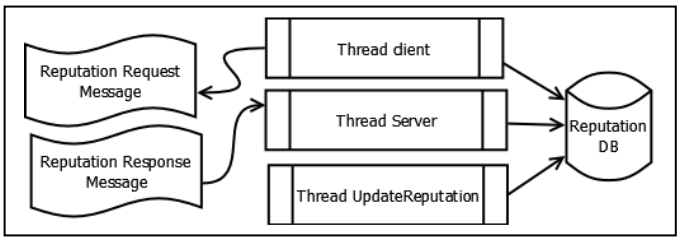

Figure 6. Basic diagram of the application PKIX-P2P

The *Thread Server* module is responsible for the publication of resources and to receive messages sent by the client module of the other CA encapsulate in JXTA Pipe Messages [2]. The *Thread Client* makes the process of resource discovery and exchange messages with the server module. The application´s main screen is shown in Figure 7.

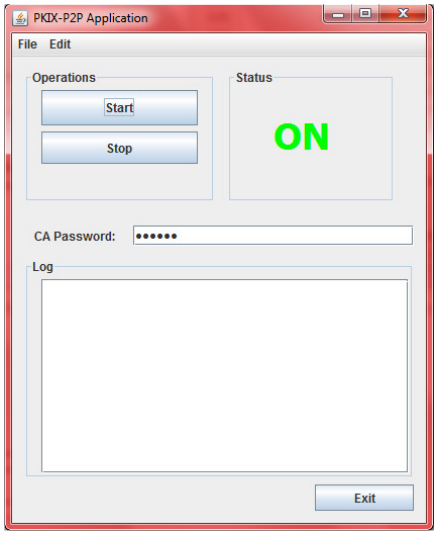

Figure 7. PKIX-P2P´s main screen.

On the main screen, the only information required is the password of the CA represented by this peer, which must be previously registered by certificate management tool. The main parameters configuration operation PKIX-P2P are defined in the settings screen, accessed via "Edit/Preferences" menu, which gives access to screen shown in Figure 8.

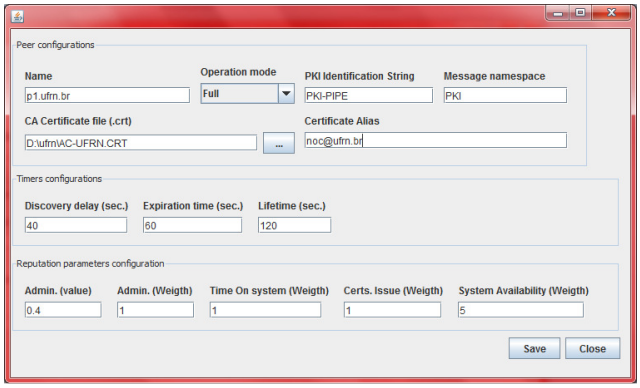

Figure 8. Configuration Screen.

This screen defines the peer configuration parameters:

- Operation mode (client only, Server only or both).
- Identification string.
- Timers.
- Weights and parameters used to calculate the reputation.

Experiments were carried out using module PKIX-P2P to evaluate the functioning, performance and network stability of the system. The experiments were performed on computers with 256 MB RAM, 800 MHz Intel processor running the operating system Windows XP SP3. Java Virtual Machine used was 1.6 [6] and LAN environment was Ethernet, which supports multicast transmission. The remaining applications running uninterrupted on three nodes, for 07h: 30m: 55s.

The graph shown in Figure 9. shows the amount of reputation messages sent/received by the CA-02 and CA-03, as counted by CA-01. For purposes of calculating the reputation, the messages sent/received by the CA-01 are not counted.

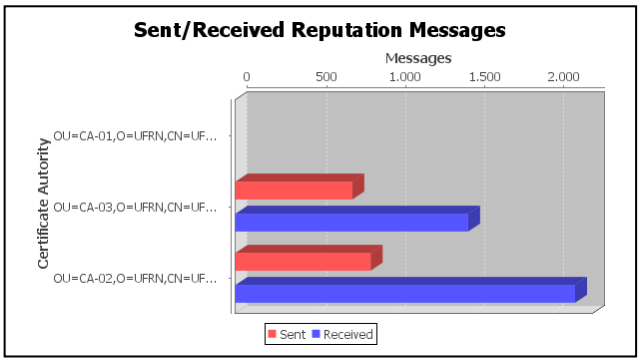

Figure 9. Amount of reputation messages sent/received counted by CA-01.

The graph shown in Figure 10. shows the reputation of the system calculated locally by the CA-01.

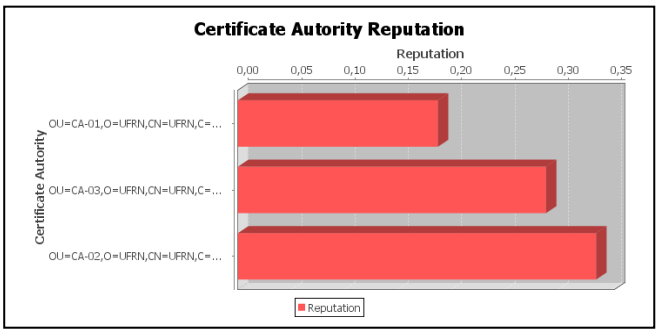

Figure 10. System reputation viewed by CA-02.

It is important to realize that the reputation of the CA-01, under his own point of view, is lower than the reputation of other CA participants in the system. This is because the messages of his own reputation CA are not counted by itself, intentionally avoiding an increase of reputation that could unbalance the system.

VI. CONCLUSION

The use of P2P technologies for the security area is seen as a major challenge because of the drastic change of paradigm. The traditional centralized applications, the concept of perimeter security and an entire culture based on client / server architecture are reasons for resistance to the adoption of P2P systems as a security solution. The certification is a reality worldwide, where the printed documents are being replaced by digital versions with legal validity. However, the certificate authorities are not fully prepared to operate in dynamic environments semi-centralized or decentralized as modern P2P networks.

Thus, we conclude that in a near future, there will be increasingly distributed systems using peer-to-peer in their operations. Thus, to grow the demand for PKI operating in this environment and that will be widely used, mainly due to its characteristics of good scalability, availability and fault tolerance.

REFERENCES

[1] Rocha, J., Domingues, M. A., Callado, A., Souto, E., Silvestre G., Kamienski, C. A., and Sadok, D. Peer-to-Peer: Computação Colaborativa na Internet. Minicursos SBRC2004 (capítulo de livro), pp. 3-46, Maio 2004.

[2] Sun Microsystems.JXTA Java™ Standard Edition v2.5: Programmers Guide. Sun Microsystems, Inc, Sep 2007.

[3] Wölfl, Thomas. Public-Key-Infrastructure Based on a Peer-to-Peer Network. In: Proceedings of 38th Hawaii International Conference on System Sciences, 2005, Hawaii, EUA. University of Regensburg, jan. 2005.

[4] Adams, C., Farrell, S.. Request for Comments 2510, Entrust Technologies. Internet X.509 Public Key Infrastructure Certificate Management Protocols, March 1999.

[5] Traversat, Bernard., Arora, Ahkil. Abdelaziz, Mohamed.,Duigou, Mike., Haywood, Carl., Hugly, Jean-Christophe., Pouyoul, Eric., Yeager, Bill. Project JXTA 2.0 Super-Peer Virtual Network. Sun Microsystems, Inc, May 2003.

[6] M. Sun, “Java Platform, Standard Edition (Java SE),” Sep 2010, on-line in http://java.sun.com/javase/downloads/index.jsp.