ISSN:0975-9646

Exchange of Peer To Peer Identity Reputation Using Cryptology conventions

S.Venkateswarlu and Md khaja zikriya

DEPT of CSE,KL University, Vaddeswaram ,Vijayawada, Andhra Pradesh, India

Abstract – **peer-to-peer architectures derives to a large extent from their ability to function, scale and self-organize in the presence of a highly transient population of nodes, network and computer failures, without the need of a central server, Peer-to-peer networks are capable of being wounded to peers who cheat, propagate malicious code, leech on the network, or simply do not cooperate. The traditional security techniques developed for the centralized distributed systems like client-server networks are insufficient for P2P networks by the virtue of their centralized nature. The absence of a central authority in a P2P network poses unique challenges for identity management in the network. These challenges include Reputation of the peers, secure directory data management, Sybil attacks, and above all, availability of Reputation data. In this paper, we present a cryptology conventions for ensuring secure and timely availability of the Reputation data of a peer to other peers at extremely low costs. The past behavior of the peer is encapsulated in its digital Reputation, and is subsequently used to predict its future actions. As a result, a peer's Reputation motivates it to cooperate and desist from malicious activities.**

Key words-**Peer-to-peer networks, distributed systems, security, identiy management, Reputations.**

I.INTRODUCTION

PEER-TO-PEER (P2P) networks are self-configuring networks with minimal or no central control. P2P networks are more vulnerable to dissemination of malicious or spurious content, malicious code, viruses, worms, and trojans than the traditional client-server networks, due to their unregulated and unmanaged nature. For example, the infamous VBS.Gnutella worm that infected the Gnutella network, stored Trojans in the host machine. The peers in the P2P network have to be discouraged from leeching on the network. It has been shown in Tragedy of Commons that a system where peers work only for selfish interests while breaking the rules decays to death. Policing these networks is extremely difficult due to the decentralized and ad hoc nature of these networks. Besides, P2P networks, like the Internet, are physically spread across geographic boundaries and hence are subject to variable laws.

The traditional mechanisms for generating trust and protecting client-server networks cannot be used for pure P2P networks. This is because the trusted central authority used in the traditional client-server networks is absent in P2P networks. Introduction of a central trusted authority like a Certificate Authority (CA) can reduce the difficulty of securing P2P networks. The major disadvantage of the centralized approach is, if the central authority turns malicious, the network will become vulnerable. In the absence of any central authority, repository, or global information, there is no silver bullet for securing P2P networks. In this paper, we investigate identity Systems for P2P networks—a more ambitious approach to protect the P2P network without using any central component, and thereby harnessing the full benefits of the P2P network. The identity of the peers are used to determine whether a peer is a malicious peer or a good peer. Once detected, the malicious peers are ostracized from the network as the good peers do not perform any transactions with the malicious peers. Expulsion of malicious peers from the network significantly reduces the volume of malicious activities. All peers in the P2P network are identified by identity certificates (aqua identity).

The Reputation of a given peer is attached to its identity. The identity certificates are generated using self-certification, and all peers maintain their own (and hence trusted) certificate authority which issues the identity certificate(s) to the peer. Each peer owns the Reputation information pertaining to all its past transactions with other peers in the network, and stores it locally. A two-party cryptology conventions not only protects the Reputation information from its owner, but also facilitates secure exchange of Reputation information between the two peers participating in a transaction. The experiments show that the proposed Reputation infrastructure not only reduces the percentage of malicious transactions in the network, but also generates significantly less network traffic as compared to other Reputation-based security solutions for P2P networks. The main contributions of this paper are:

1. A self-certification-based identity system protected by cryptology blind identity mechanisms.
2. A light weight and simple Reputation model.
3. An attack resistant cryptology protocol for generation of authentic global Reputation information of a peer.
4. The latency associated with a file replication in a P2P system. The peers accounts for the file the size distribution.
5. The search time.
6. Load distribution at peers.

II.RELATED WORK

Structured and Unstructured P2P Networks: P2P networks can be categorized into structured and unstructured P2P networks. The proposed system can be used on top of both structured and unstructured P2P networks. In structured networks, the location of the data is a function of data itself or its metadata. As a result, the search space is constrained by the metadata. The overlay networks like Chord, CAN, and PASTRY are structured networks and as a result the search in these networks (Chord is O(log N) where N is the number of nodes in the network) is much more efficient than in purely unstructured (without any super nodes) P2P networks. Moreover, in structured networks, all the nodes know the fundamental structure of the network and hence can prune their search to the relevant nodes. The unstructured P2P networks do not have a well known architecture. In unstructured networks, there is no relationship between the data or metadata and its location. As a result search is of the order of O(N) in these networks, where N is the number of nodes (each node will receive a query message at least once).

The identity schemes proposed in this paper are independent of the structure of the network. It assumes that a search function is available and does not put any constraint on the implementation of the search function. As a result, the proposed scheme is equally useful in both the unstructured and the structured networks. The knowledge of the structure of the network can be used for optimizing the algorithm. We do not assume any such knowledge in this paper.

R-CHAIN

It is lightweight reputation management system R-Chain where each peer maintains its own transaction history as the reputation. Each transaction in R-Chain involves two equal parties and use file downloading as the example. Each transaction will result in a transaction record (TR) as the proof of its existence R-chain minimizes the maintenance and retrieval cost by maintaining the transaction history on the owner node.

SYBIL ATTACK

If a single faulty entity in a P2P system can present in multiple identities it can control a substantial fraction of the system thereby undermining this redundancy. Sybil attacks are always possible except under extreme and unrealistic assumptions of resource parity and coordination among entities.

IDENTITY MODELS

Resnick et al.defines the identity system as “a system that collects, distributes, and aggregates Reputation about consumer’s past behavior.” The authors outline the problems in eliciting, distributing, and aggregating Reputation. Resnick et al. explain the problem of pseudospoofing in. Pseudospoofing is the use of multiple pseudonyms in a system by the same real-life entity. The disadvantage is that any entity can discard a handle or a pseudonym with which a bad Reputation is associated and join the system as a new user, under a new pseudonym. This can possibly nullify the usefulness of a Reputation system, which assigns Reputations to handles. The authors also advocate that the newcomers should pay their dues in order to mitigate the effect of pseudo spoofing. In other words, the newcomers should not only use the services of the system but should also contribute to the system as per the system guidelines. Peer Trust allocates the Reputation information to a certain node on the network for storage, by using hash functions. Any peer looking for the Reputation of another peer uses the search mechanism of the underlying network to search for the information. The authors of Peer Trust argue that trust models based solely on Reputation from other peers in the community are ineffective and inaccurate. The authors recommend the “degree of satisfaction” of the peer from previous transactions and the number of transactions a peer performs in the system should be accounted for before calculating the Reputation of the recommended peer.

Abdul-Rahman and Hailes have proposed another trust model with corresponding metrics. They argue that Bayesian probability may not be the best metric for representing degree of trust, because probability is inherently transitive while trust is not. In addition, the authors provide methods for combining recommendations and use the context of recommendations and recommender weights to evaluate the Reputations from recommendations.

Aberer and Despotovic have proposed completely distributed solution for trust management over the P-Grid peer-to-peer network. They store Reputation data in the form of a binary search tree, over the network. Any agent looking for the recommendation data of another agent searches the P2P network and computes the Reputation from the recommendations received. Chen and Singh and Schein et al. also provide trust models, similar to those mentioned above.

Dellarocas has enumerated the design challenges in the online reporting systems. Dellarocas surveys online Reputation, reporting mechanisms, and the corresponding issues. In addition, the author provides a good overview of recommendation repositories, professional rating sites, collaborative filtering systems, and regression approaches. Delarosa also enumerates the attacks on Reputation systems and techniques for foiling those attacks. The attacks that can be inflicted on the Reputation systems are ballot stuffing and bad mouthing. In ballot stuffing, a peer receives a large number of (false) positive recommendations from its friends, to raise its own Reputation. Bad mouthing implies issuing a large number of negative recommendations for a specific peer. The author advocates that the problems of negative and positive discrimination can be solved by maintaining anonymity of requesters

III. Reputation system

Threat Model

A Gnutella-like network has a power-law topology and supports Insert and Search methods. The peers follow predefined Join & Leave protocols. The peers are connected with insecure communication channels. As the peers are likely to have conflicting interests, a source of motivation is needed to reduce the number of lechers. Lechers are peers who derive benefit from the system without contributing to the system. The rogue peers can also spread malware in the network (when other peers download content from them). Finally, peers need a mechanism to judge the quality of the content before making Go/No-Go decision in transactions and thereby develop trust relationships with other peers. A perfect Reputation system can provide the means to achieve the above goals. Any Reputation system is vulnerable to ballot stuffing and bad mouthing as described.

Self-Certification

In order to participate in the identity system, a peer needs to have a handle. The identity of a peer is associated with its handle. This handle is commonly termed as the “identity” of the peer even though it may not “identify” a peer, i.e., it may not lead to the real-life identity of the peer. A peer receives a recommendation for each transaction performed by it, and all of its recommendations are accumulated together for calculation of the identity of a given peer. In a centralized system, a trusted authority would have issued these identity certificates. In a decentralized Reputation system, self-certification splits the trusted entity among the peers and enables them to generate their own identities. Each peer runs its own CA that issues the identity certificate(s) to the peer. All the certificates used in self certification are similar to SDSI certificates [6]. The Reputation of a peer is associated with its identity and the Reputation of a CA is the accumulated Reputation of the identities. Self-certification obviates the centralized trusted entity needed for issuing identities in a centralized system. Peers using self-certified identities remain pseudononymous in the system as there is no way to map the identity of a peer in the system to its real-life identity. Although anonymity or at least pseudonym is extremely desirable in P2P networks, in a Reputation system it is a double edge sword. In the absence of any mapping between multiple identities and their owner (peer), the system is vulnerable to Sybil attack or Liar farms. A malicious peer can use self-certification to generate a large number of identities and thereby raise the Reputation of one of its identities by performing false transactions with other identities. The malicious peer does not even need to collude with other distinct peers to raise its Reputation, but only needs to generate a set of identities for itself. Such a large set of identities managed by one peer is called an identity farm. The set of identities that issue false recommendations is called a liar farm. This attack belongs to the class of attacks termed sybil attacks. In simple words, a peer having an identity farm is equally capable of subverting a Reputation system as a peer that has colluded with a large number of other peers. An identity farm can be countered if, either a peer is restricted to one identity or all the identities of a peer can be mapped back to the peer. A peer can be restricted to one identity by mapping its identity to its real-life identity and thereby sacrificing anonymity, or by making the identity generation extremely resource intensive such that the peer cannot afford to generate multiple identities. Identity generation can be made resource intensive by using traditional micro payment methods, although the resource restrictions are likely to have a varied impact depending on each peer's resourcefulness. In self-certification, we use a combination of both approaches. Each peer's CA can generate multiple identities. The recommendations received for a peer's identity from different identities of other peers, signed by the other peer's CA(s), are identified as signed by the same CA, and are averaged to counter the liar farms. In a transaction, the requester averages all the recommendations of the provider by CAs of the provider's past recommenders. Hence, all the past recommendations owned by the provider carry equal weight but they get averaged. Finally, it adds the averages of each CA to calculate the Reputation of the provider identity. Hence, a peer cannot use its own identities (all generated by the same CA) to recommend its other identities. A more determined malicious peer might start multiple CAs and generate multiple groups of identities. In order to counter a rogue peer having multiple CAs, the peers are divided into groups based on different criteria such that a peer cannot become a part of multiple groups. Each peer obtains its group certificate from the appropriate authority and attaches it to its CA. The certificate of a group authority is publicly accessible by any node inside or outside the group. The peer sends its blinded credentials to the group authority and the authority verifies the credentials and signs the group certificate. The authority remains stateless, i.e., it does not maintain any information to correlate a certificate with the peer. Unlike the traditional CA or distributed CA-based approaches, grouping of peers preserves the anonymity of the peers; when combined with self-certification it curtails the possibility of a Sybil attack. In contrast to the traditional CA-based approach, neither the group authority nor the transacting peers can establish the identity of the peer. In addition, certificate revocations are not needed in the group-based approach as the group authority only vouches for the real-life existence of the peer, unlike the traditional certificate-based approaches where various certificate attributes are attested by the authority and necessitate revocation if any of those attributes mutate in time. If a highly reputed identity is compromised, its misuse would be self-destructive as its Reputation will go down if misused.

In order to participate in the Reputation system, a peer needs to have a handle. The Reputation of the peer is associated with its handle. This handle is commonly named as the 'identity' of the peer even though it may not "identify" a peer. A peer receives a recommendation for each transaction performed b it, and all of its recommendations are accumulated together to the calculation of the Feedback of the peer. A malicious peer can use self-certification to generate a large number of identities and thereby raise the Reputation of one of its identities by performing false transactions with other identities. The malicious peer does not even need to collude with other peers to raise its Reputation, but only needs to generate a set of identities for itself. Such a large set of identities managed by one peer is called identity farm. The set of identities that issue false recommendation is called a liar farm. more determined malicious peer might start multiple CAs and generate multiple groups of identities. In order to counter a rogue peer having multiple CAs, the peers are divided into groups based on different criteria such that a peer cannot become a part of multiple groups.

The peer is denoted by P while the authority is denoted by A. Here P→A : X denotes that the peer (P) sends a message X to the authority (A). The symbol PK2 represents the private key of the peer P and PK1 represents the public key of the peer P. EK(T) represents encryption of the phrase (T) with key K, while EBK(X) represents blinding phrase X with key K.

P→A:B1={EBka(IAlicer)},IAlicer

The peer Alice generates a BLINDING KEY, Ka and another identity for herself (IAlicer). Alice cannot be identified from her identity (IAlicer). Subsequently, she blinds her identity (IAlicer) with the blinding key Ka. B1 represents the blinded identity. Alice sends B1 to the authority with her real identity that proves her membership to a group.

A→P:B2=EpAuthorityK2{B1=EBka(IAlicer)}

The authority signs the blinded identity, B1 and sends it (B2) back to the peer. The peer unblinds the signed identity and extracts the identity authorized by the authority EPAuthorityK2 {IAlicer}.
P:EPAuthorityk2{ IAlicer}= EBka{B2}}

The fundamental assumption in the group-based approach is that in a P2P network, peers would be more interested in the ranks of the prospective providers than in the absolute value of the Feedbacks. The simulations show that this approach changes the absolute Reputations of peers considerably but it has only a minimal impact on the relative ranks of the peers. This approach is inspired from the Google page rank concept in which the pages in proximity of each other do not contribute as much to the page rank of the target page as compared to pages at a distance [34]. The relative ranks do not stop the peers from setting thresholds. The thresholds can be based on ranks. Setting thresholds based on absolute values has very limited utility. Google uses ranks rather than the absolute numbers of links pointing to/from pages. It is well evident from the Google example that rank-based mechanisms are scalable. It can be argued that there might be some systems where absolute values are needed. This paper does not consider that case as use of absolute values needs more system context specific information that is outside the focus of this paper.

REPUTATION MODEL

Once the peer is obtained its identity, it joins in the P2P network using join method of that network. Peer searches for one or more files using the search method provided by the network. If peer have the response corresponding to that the particular peer is responded. The number of peer who offers a particular file is denoted by RANGE. The requester selects the highest Reputation peer from the list then initiates the protocol sends the recommendation cryptographic protocol. Depending on the verification MIN_RECOMMENTATION and MAX_RECOMMANDATION is given to the provider. The recommendation have constrains that one recommendation completely nulls or improves the Reputation of the any peer in the network. The proposed model is independent of the topology of the P2P.

REPUTATION EXCHANGE PROTOCOL

Once the requester has selected the provider with the highest Reputation, it initiates the Reputation exchange protocol with the provider. In the Reputation exchange protocol, the requester is denoted by R while the provider is denoted by P. Here R→P: X denotes that the requester (R) sends a message X to the provider (P). The symbol Pk2 represents the private key of the peer P and Pk1 represents the public key of the peer P. EK (T) represents the encryption of the phrase (T) with key K, H(λ) denotes one way of hash of the value of the λ. This protocol assumes only insert & search methods are available and they are resilient to peers that may not follow the recommended join & leave protocol of the network.

Step1:R→P:RTS&IDC

The requester sends a REQUEST FOR TRANSACTION (RTS) and its own IDENTITY CERTIFICATE (IDR) to the provider. The provider needs the identity certificate of the requester as the provider has to show it to the future requesters in Step 7.

Step2:P→R:IDC&TID&EPk2(H(TID||RTS))

The provider sends its own IDENTITY CERTIFICATE (IDP), the CURRENT TRANSACTION ID (TID) and the signed TID, EPk2(H(TID||RTS). The signed TID is needed to ensure that the provider does not use the same transaction id again. In the end of the protocol, this signed TID is signed by the requester also and stored into the network where it will be accessible to other peers.

Step3:R:LTID=Max(Search(PK1||TID))

The requester obtains the value of the LAST TRANSACTION ID (LTID) that was used by the provider, from the network. The requester concatenates the public key of the provider with the string TID and performs the search. Any Peer having the TID for the provider replies back with the TID and the requester selects the highest TID out of all the TIDs received. The highest TID becomes the LTID. It is possible that the provider might collude with the peer who stores its last LTID and change the LTID. As the LTID and related information would be signed by the requester, the provider cannot play foul.

Step4:R:IF(LTID≥TID)GO TO Step12

If the value of the LTID found by the requester from the network is greater than or same as the TID offered by the provider, it implies that the provider has used the TID in some other transaction. Hence, it is trying to get another recommendation for the same transaction number (TID). The requester suspects foul play and jumps to Step 12.

Step5:R→P:Past Recommendation Request & r

. If the check in Step 4 succeeds, i.e., the requester is sure that the provider is not using the same transaction number,

it requests the provider for its previous recommendations. In other words, if the current transaction is the Nth transaction for the provider, the requester makes a request for N -1th;N -2th and so on

recommendations till N – rth recommendation where r is less than N. The value of r is decided by the requester and it is directly proportional to the requester's stake in the transaction
Step6: P→R:CHAIN,E Pk2(CHAIN)

CHAIN = ({RECN-1||EZ N-1 K2(H(REC N-1)}||

{RECN-2||EZ N-2K2(H(REC N-2, REC N-1))}||…

{RECN-4||EZ N-4K2(H(REC N-r, REC N-r-1))})

The provider sends its past recommendations (RECN-1;RECN-2 . . .RECN-3) which were provided by peer (ZN-1; ZN-2; . . .ZN-3). The provider signs the CHAIN so that the requester can hold the provider accountable for the chain. As the recommendations have been signed by the previous requesters, the provider could not have maliciously changed them. If the requester (say Zl) has signed both the (lth) and the previous (l-1th) recommendation using its private key ZK2, as EZnK2 (H(REC N-3, REC N-(l-1)) , there is no way a provider can modify the CHAIN. In other words, the provider cannot simply take away a bad recommendation and put in a good recommendation in order to increase
its Feedback.

Step7:R:Result=Verify (RECN-1, RECN-2, RECN-r)

If Result !=Verified GO TO Step 12

The requester verifies the CHAIN by simple public key cryptography. If it has the certificates of all the peers with whom the provider has interacted in the past, the verification is simple. In the case it does not have the required certificates; it obtains the certificates from the provider itself. The provider had obtained its requester's certificate in Step 1. In addition, the requester checks for liar farms as mentioned in paragraph 2 of Section 3.2. If the verification fails the requester jumps to Step 12
Step8: P→R:File or Service

The provider provides the service or the file as per the requirement mentioned during the search performed for the providers.
Step9: R→P:B1 =
EBka(REC||TID||E{HRK2(REC,||TID)})

Once the requester has received a service, it generates a BLINDING KEY, Ka. The requester concatenates the RECOMMENDATION (REC) and the TRANSACTION ID (TID) it had received in Step 2 and signs it. Subsequently, it blinds the signed recommendation with the blinding key, Ka. The recommendation is blinded in order to make the provider commit to the recommendation received before it sees the value of the recommendation such that it does not disown the recommendation if it is low. The provider receives the blinded recommendation from the requester. The blinded recommendation is also signed by the requester. The blinded recommendation contains the Chain that the provider can subsequently use to validate its Reputation to another requester.
Step10:

a. P→R:B1||EPK2(H(B1),nonce),nonce
b. R→P:Ka

The provider cannot see the recommendation but it signs the recommendation and sends the NONCE and the signed
recommendation back to the requester. The requester verifies the signature and then sends the blinding key Ka to the provider which can unblind the string received in Step 10a and checks its recommendation.
Step11:Insert

(IDR,{CHAIN||TID||ERK2{H(REC)||H(TID)}})

The requester signs: the recommendation that was given to the provider (REC), the transaction id (TID), and its own identity certificate and stores it in the network using the Insert method of the P2P network. This completes the transaction.
Step12: Step 12 explains the steps a requester executes when it expects foul play:
ABORT PROTOCOL
R:Insert(IDR,{CHAIN||TID||ERK2{H(CHAIN)||H(TID)}})

If the verification in Step 7 fails, the requester takes the CHAIN that was signed by the provider and the Transaction Id (TID), signs it and uses the INSERT method of the network to insert the chain and its own identity certificate into the network. As a result, any subsequent requester will be able to see failed verification attempt and will assume a MIN RECOMMENDATION recommendation for that TID for the provider. The requester cannot insert fake recommendations into the network because it has to include the TID signed by the provider. If the requester reaches Step 12 from Step 4. It will request for the Chain from the Provider and subsequently will perform
R:Insert(IDR,{CHAIN||TID||ENRK2{H(TID||RTS))}})

IV.CONCLUSION

This paper presents self-certification, an identity management mechanism, reputation model, and a cryptology conventions that facilitates generation of global reputation data in a P2P network, in order to expedite detection of rogues. A Reputation system for peer-to-peer networks can be thwarted by a consortium of malicious nodes. Such a group can maliciously raise the reputation of one or more members of the group. There is no known method to protect a Reputation system against liar farms and the absence of a third trusted party makes the problem of liar farms even more difficult.

The self-certification-based identity generation mechanism reduces the threat of liar farms by binding the network identity of a peer to his real-life identity while still providing him anonymity. The Identity mechanism is based on the fundamental that the ranks of the peers are more relevant than the absolute value of their reputation. The cost of this security is the difference in the ranks of the providers because of the use of the proposed mechanism.

The global reputation data are protected against any malicious modification by the third party peer and are immune to any malicious modifications by their owner. The proposed protocol reduces the number of malicious transactions and consumes less bandwidth per transaction than the other Reputation systems proposed in its category. It also handles the problem of highly erratic availability pattern of the peers in P2P networks.

ACKNOWLEDGEMENTS

We are greatly delighted to place my most profound appreciation to Dr.K.Satayanarayana Chancellor of K.L.University, Dr.K.Raja Sekhara Rao Principal, S.Venkateswarlu Head of the Department ,and Dr Subramanyam in charge for M.Tech under their guidance and encouragement and kindness in giving us the opportunity to carry out the paper .Their pleasure nature,directions,concerns towards us and their readiness to share ideas enthused us and rejuvenated our efforts towards our goal. We also thank the anonymous comments references of this paper for their valuable

REFERENCES

[1] J. Douceure, “The Sybil Attack”, *Proc, IPTPS ’02 Workshop,* 2002.

[2] A. Abdul-Rahman and S. Hailes, “Supporting Trust in Virtual Communities”, *Proc. Hawaii Intl Conf. System Sciences*, Jan 2000.

[3] M. Gupta, P. Judge, and M. Ammar, “A Reputation System for Peer-to-Peer Networks,”*Proc. 13th Int’l Workshop Network and Operating systems Support for Digital Audio and Video (NOSSDAV)* , 2003.

[4] L. Liu, S. Zhang, K.D. Ryu, and P. Dasgupta, “R-Chain: A self Maintained Reputation Management System in p2p Networks,” *Broc.17th Int’l Conf. Parallel and Distributed Computing Systems* (PDCS), Nov. 2004.

[5] Prashant Dewan and Partha Dasgupta,” P2P reputation Exchange Protocol Using Distributed and Decentralized Recommandation Chains, *IEEE Transaction on Knowledge and Data Engineering vol 22, No.7,* July 2010.

[6] H.Garett, “Tragedy of Commons,” *Science*, vol.162, pp.1243-1248, 1968.

[7] R.Zhou, K.Hwang, and M. Cai, “ Gossiptrust for Fast reputation Aggregation in Peer-to-Peer” *IEEE Trans. Knowledge and Data Eng.,*vol.20,no. 9, pp. 1282-1295, Aug. 2008.

[8] L.Xiong and L. Liu, “PeerTrust: Supporting Reputation-Based Trust in Peer-to-PeerCommunities”, *IEEE Trans. Knowledge and Data Eng.,* vol.16,no. 7, pp. 843-857, July 2004.

[9] B.C.Ooi, C.Y.Kiau, and K.Tan, “Managing Trust in Peer-to-Peer Systems Using Reputation-Based Techniques,” Proc .Fourth Int’l Conf.Web Age Information Management, Aug. 2003.

[10] G. Networks, “Groove Networks,” http://www.groove.net/products/workspace/securitypdf.gtml, 2009.