Volume 2, Issue 5, May 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Experimental Analysis of P2P Reputation Management Using Distributed Identities and Decentralized Recommendation Chains

Yallamati Prakasarao[1],
M.Tech Student, Department Of CSE, KIET, Kakinada
E-Mail: prakashlnr@gmail.com

A.VeeraSwamy[2],
Research Scholar, (VELTECH Dr.RR & Dr.SR TECHNICAL UNIVERSITY)
Asst.Professor, Department of IT
VRS & YRN COLLEGE OF ENGG &TECH, Chirala.
E-Mail: ammisetty.veeraswamy@gmail.com

K.Ramesh[3],
Asst.Professor,
Department Of CSE, KIET, Kakinada.

K. Ravi Kumar[4],
Head of Department,
Department Of CSE, KIET, Kakinada.

***Abstract*— 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. By 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. The reputations 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 cryptographic protocol 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.**

***Keywords*— Peer-to-peer networks, Blow fish algorithm, reputation, distributed systems, security.**

I. INTRODUCTION

A computer network, often simply referred to as a network, is a collection of computers and devices interconnected by communications channels that facilitate communications and allows sharing of resources and information among interconnected devices. Computer network is broadly classified in to two types.

1. Client Server Model
2. Peer-to-peer Model

1.1 Client and Server Model

A client server network is defined as specific type of online network comprised of a single central computer acting as a server that directs multiple other computers, which is referred to as the clients. By accessing the server, clients are then able to reach shared files and information saved on the serving computer. Further, client server networks are very similar in nature to peer to peer networks with the exception that it is only the server that can initiate a particular transaction. The client–server model of computing is a distributed application that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

A server device typically stores files and databases including more complex applications like Web sites. Server devices often feature higher-powered central processors, more memory, and larger disk drives than clients.

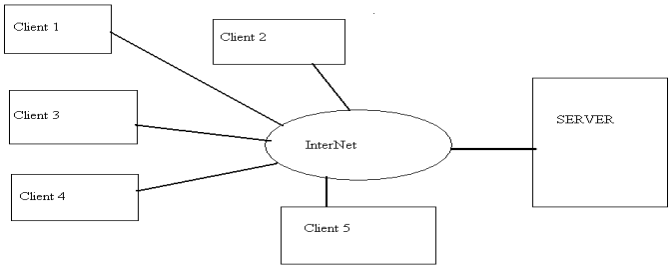


Fig 1.1 Client server architecture

A. Advantages of Client-Server Model:

- Centralization - access, resources, and data security are controlled through the server
- Scalability - any element can be upgraded when needed
- Flexibility - new technology can be easily integrated into the system Interoperability - all components (clients, network, servers) work together

B. Disadvantages of client-Server Model:

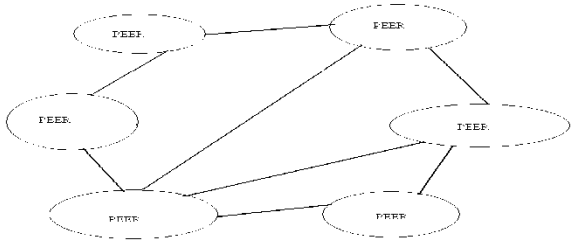


Fig 1.2 Peer-peer Network Model

The peer-to-peer application structure was popularized by file sharing systems like Napster. The concept has inspired new structures and philosophies in many areas of human interaction. Peer-to-peer networking is not restricted to technology, but covers also social processes with a peer-to-peer dynamic.

Peer-to-Peer (P2P) networking is a fairly popular concept. Networks such as Bit Torrent make it easy for people to find what they want and share what they have. The concept of sharing seems benign enough. If I have something you want and you have something I want, why shouldn't we share? For one thing, sharing files on your computer with anonymous and unknown users on the general public Internet goes against many of the basic principles of securing your computer.

A peer-to-peer (P2P) network is created when two or more PCs are connected and share resources without going through a separate server computer. A P2P network can be an ad hoc connection—a couple of computers connected via a Universal Serial Bus to transfer files. A P2P network also can be a permanent infrastructure that links half-dozen computers in a small office over copper wires. Or a P2P network can be a network on a much grander scale in which special protocols and applications set up direct relationships among users over the Internet.

C. P2P FILE SHARING NETWORKS

When most people hear the term "P2P", they think not of traditional peer networks, but rather peer to peer file sharing over the Internet. P2P file sharing systems have become the single most popular class of Internet applications in this decade. A P2P network implements

- Dependability - when the server goes down, operations cease
- Lack of mature tools - it is a relatively new technology and needed tools are lacking
- Lack of scalability - network operating systems (Novell Netware, Windows NT Server) is not very scalable.
- Higher than anticipated costs
- Can cause network congestion.

1.2 INTRODUCTION TO PEER-TO-PEER NETWORK MODEL

Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes. Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client–server model where only servers supply (send), and clients consume (receive).
search and data transfer protocols above the Internet Protocol. To access a P2P network, users simply download and install a suitable P2P client application. Numerous P2P networks and P2P software applications exist. Some P2P applications work only with one P2P network, while others operate cross-network. Likewise, some P2P networks support only one application, while others support multiple applications.

II. LITERATURE SURVEY

A. Rowstron and P. Druschel [1] had given information about the Pastry, a scalable, distributed object location and routing substrate for wide-area peer-to-peer applications. Pastry performs application-level routing and object location in a potentially very large overlay network of nodes connected via the Internet. It can be used to support a variety of peer-to-peer applications, including global data storage, data sharing, and group communication and naming.

Each node in the Pastry network has a unique identifier (nodeId). When presented with a message and a key, a Pastry node efficiently routes the message to the node with a nodeId that is numerically closest to the key, among all currently live Pastry nodes. Each Pastry node keeps track of its immediate neighbors in the nodeId space, and notifies applications of new node arrivals, node failures and recoveries. Pastry takes into account network locality; it seeks to minimize the distance messages travel, according to a scalar proximity metric like the number of IP routing hops. Pastry is completely decentralized, scalable, and self-organizing; it automatically adapts to the arrival, departure and failure of nodes

L. Zhou, F. Schneider, and R. Renesse [2], in a public key infrastructure, a certificate specifies a binding between a name and a public key or other attributes. Over time, public keys and attributes can change: a private key might be compromised, leading to selection of a new public key, for example. The old binding and any certificate that specifies that binding then become invalid. A certification authority (CA) attests to the validity of bindings by issuing digitally signed certificates that specify these bindings and by providing a means for clients to check the validity of certificates. With an online CA, principals can check the validity of certificates just before using them.

COCA (Cornell Online Certification Authority), the subject of this article, is such an online CA. COCA employs replication to achieve availability and employs proactive recovery with threshold cryptography for digitally signing certificates in a way that defends against mobile adversaries [Ostrovsky and Yung 1991] which attack, compromise, and control one replica for a limited period of time before moving on to another. In that, the system is not novel. What distinguishesCOCA are its qualitatively weaker assumptions about communication links and execution timing. Many denial-of-service attacks succeed by invalidating stronger communication and execution-timing assumptions; in making weaker assumptions, COCA is less vulnerable to these attacks.

New proactive recovery protocols had to be developed for execution in this relatively unconstrained and more realistic environment. Moreover, because implementing agreement is problematic in the absence of execution-timing assumptions [Fischer et al. 1985], in so doing, COCA is the first to tackle the problems associated with integrating threshold cryptography and Byzantine quorum systems. Thus, beyond its intrinsic utility for public key infrastructures, COCA has pedagogical value as a vehicle for understanding how to combine mechanisms for supporting fault-tolerance and security properties. Besides its weak assumptions, a variety of traditional means for combating denial-of-service attacks is used by COCA: (i) processing only those requests that satisfy authorization checks, (ii) grouping requests into classes and multiplexing resources so that demands from one class cannot have an impact on processing of requests from another, have an impact on as well as (iii) caching results of expensive cryptographic operations. And although resource-clogging denial-of-service attacks certainly remain possible, experiments demonstrate that launching a successful attack against COCA is harder with these mechanisms in place.

III. INTRODUCTION TO THE REPUTATION SYSTEM

It has been proven useful in online decision making processes and risk management. In P2P networks where peers rarely know each other reputation data helps peers to trust each other. Most reputation systems have a centralized control and concentrate on reputation given to users. It is difficult to implement such a centralized control in decentralized P2P networks. Using peer reputation reduces malicious transactions by filtering out malicious peers. Combining peer reputation with file reputation increases the efficiency of the reputation system and consequently increases the efficiency of a P2P network with a reduced number of malicious peers, malicious transactions and malicious files.

Reputation systems that use only peer reputation can be spoiled by malicious users. Peer reputation systems could bring more security and efficiency to the reputation systems. We describe the basic idea of two reputation systems. The first one: A Peer-to-peer reputation in decentralized environments (PRIDE) is a reputation system that facilitates peer reputation and transaction reputation. In this reputation system peers use Simple Distributed Security Infrastructure (SDSI) for self-certification. Using self-certificates does not prevent peers from creating multiple identities. A set of peers that create multiple identities and give each other good reputation values is called a 'liar farm' Countering 'liar farms' is discussed in. Using PRIDE, a requesting peer (requester) sends a search query to the network and receives a result query that includes peer reputation data of the providing peers.

The requesting peer chooses a providing peer (provider) with the best reputation. The provider then generates a new transaction id (TID). The TID is generated using the last transaction id as a parameter to the one-way function. The provider sends the new TID to the requester. The requester checks if the new TID has been used before in any transactions in the network. If that is the case, the requester checks if this TID has received any recommendations. If the requester is satisfied with the provider's reputation data and TID recommendations it initiates the transaction. Once the transaction is complete the requester sends a signed peer reputation recommendation to the provider. Accordingly, the requester signs the TID and will provide it to other peers upon a request in the network.

IV. SALIENT FEATURES OF THE PROTOCOL

The main features of the protocol are as follows

1. Legitimate global reputation information w.r.t. a given provider is available to all peers at one place (with the provider itself). The requester does not have to initiate multiple search requests in the network in order to collect the recommendations received by the provider in the past. It only has to issue one search request to retrieve the last transaction information of the provider and it can verify all the recommendations of the provider. This not only reduces the turnaround time of the transaction butalso saves considerable volume of resources.
2. The provider is accountable for all its past transactions. It cannot maliciously meddle with its transaction history by adding or deleting any recommendation because the recommendations are chained in a sequence and signed by the past requesters. The provider cannot change any recommendation because they are digitally signed by the requesters.
3. As the global information of the provider is stored by the provider itself, this protocol is not affected by errati availability of past recommenders or any other peer in the network. As long as the requester and the provider are connected to the network, the transaction can be completed fruitfully. Even if one of them leaves the network it can come back and complete the transaction. In short, it handles the problem of irregular availability of the peers in the network, which is one of the major problems in P2P networks.
4. This protocol cannot stop a requester from giving a "bad" recommendation to the provider even if the latter provides a legitimate file. This protocol does not stop bad mouthing nor does it prevent ballot stuffing. The proposed scheme necessitates collusion among a large number of peers for ballot stuffing or bad mouthing to work, and hence it is unlikely to be successful in the proposed system although it is not impossible. By using the Blowfish algorithm we can perform data encryption and decryption so that we can achieve the high security.

V. BLOWFISH ALGORITHMS

The data transformation process for Pocket Brief uses the Blowfish Algorithm for Encryption and Decryption, respectively. The details and working of the algorithm are given below. Blowfish is a symmetric block cipher that can be effectively used for encryption and Safeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data.

Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key

can be any length up to 448 bits. Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors. Blowfish is a variable-length key block cipher. It is suitable for applications where the key does not change often, like a communications link or an automatic file encryption. It is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches.

The Blowfish Encryption/Decryption Algorithm:

- Manipulates data in large blocks
- Has a 64-bit block size
- Has a scalable key, from 32 bits to at least 256 bits.

Blowfish is a block cipher that encrypts data in 8-byte blocks. The algorithm consists of two parts: a key-expansion part and a data-encryption part. Key expansion converts a variable-length key of at most 56 bytes (448 bits) into several subkey arrays totaling 4168 bytes.

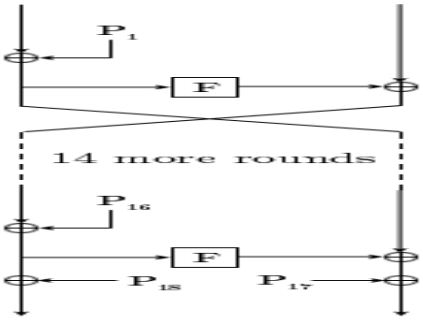


Fig 5.1 Feistel structure of blow fish algorithm

Blowfish has 16 rounds. Each round consists of a key-dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round. Blowfish uses a large number of sub keys.

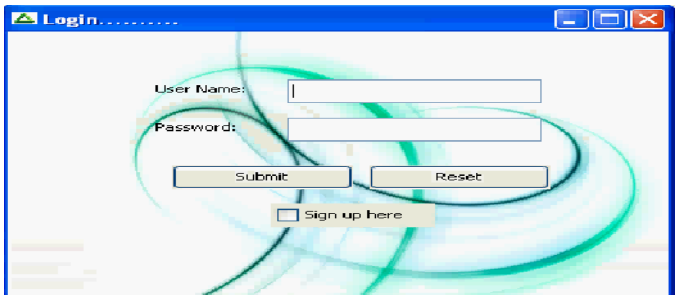
These keys must be recomputed before any data encryption or decryption. The P-array consists of 18 32-bit sub keys: P1, P2... P18. There are also four 32-bit S-boxes with 256 entries each: S1,0, S1,1,..., S1,255; S2,0, S2,1,..,, S2,255; S3,0, S3,1,..., S3,255; S4,0, S4,1,..,, S4,255.

- Blowfish has 16 rounds.
- The input is a 64-bit data element, x.
- Divide x into two 32-bit halves: xL, xR.
- Then, for i = 1 to 16 rounds:
- xL = xL XOR Pi
  xR = F(xL) XOR xR
  Swap xL and xR
- After the sixteenth round, swap xL and xR again to undo the last swap.
- Then, xR = xR XOR P17 and xL = xL XOR P18.
- Finally, recombine xL and xR to get the cipher text.
- Function F looks like this: Divide xL into four eight-bit quarters: a, b, c, and d. Then, F(xL) = ((S1, a + S2,b mod 232) XOR S3,c) + S4,d mod 232

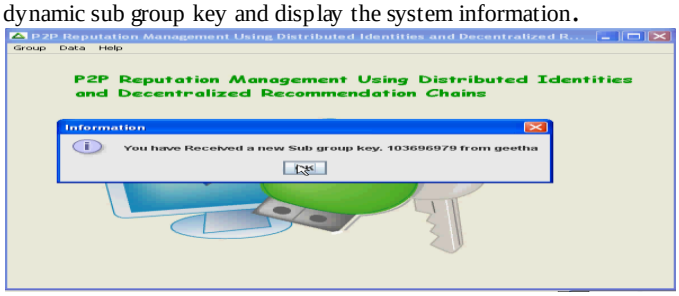
Decryption is exactly the same as encryption, except that P1, P2... P18 are used in the reverse order.

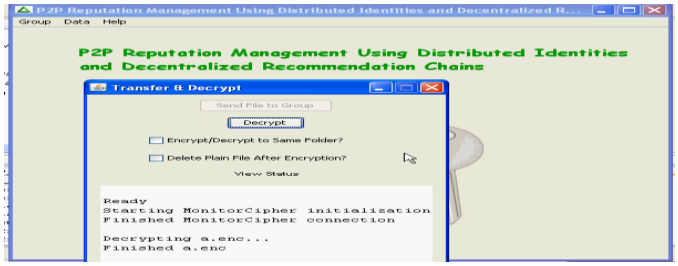
VI. EXPERIMENTS AND RESULTS

This fig helps us as to login the administrator for selecting the destination to transfer the data.



This figure helps us as after joining the peer and it will generate the dynamic sub group key and display the system information.

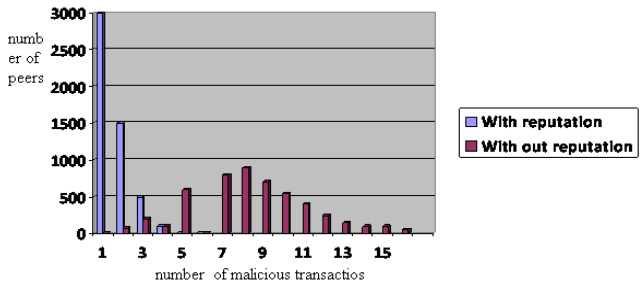




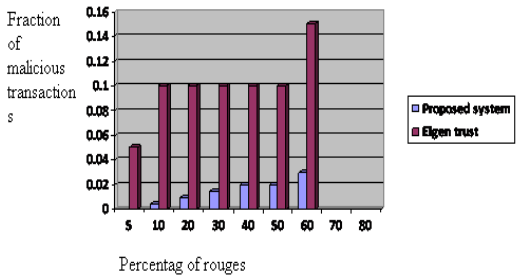
This figure helps us as after received the file from the source and that file only to decryption is performed and display the status of the process

VII. PERFORMANCE ANALYSIS

The total number of malicious transactions increased considerably with an increase in the number of transactions when the proposed model was not used but are more or less constant when the proposed model was used. In the presence of an increasing number of rogues when the total number of transactions is constant the rate of increase in the number of malicious transactions was much less when reputations were used Variation in total number of malicious transactions. The peers versus the malicious transactions.



The reputations were not used, this mean the number of transactions drastically reduced, and when the reputation model is used we concluded that the proposed model reduces the number of malicious transaction from the perspective of the network.The proposed system is more effective in reducing the number of "inauthentic downloads" or "malicious transactions." In a network where 60 percent nodes are malicious, the proposed system reduces the number of malicious transactions from 15 percent (in Eigen Trust) to 2 percent.



Percentage of rouges vs. the malicious transactions

VIII. CONCLUSION & FEATURE ENHANCEMENT

This paper presents self-certification, an identity management mechanism, reputation model, and a cryptographic protocol 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. Currently, the reputation of the provider is considered and the reputation of the requester is ignored. This system can be extended to encapsulate the reputations of both the provider and the requester. In addition, instead of generic number values, the reputation values can be modified in accordance with the context of the reputation.

REFERENCES

[1] H. Garett, "Tragedy of Commons," Science, vol. 162, pp. 1243-1248,1968.

[2] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M.F. Kaashoek, F.Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," Proc. ACM SIGCOMM, pp. 149-160, Aug. 2002.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A Scalable Content-Addressable Network," SIGCOMM Computer Comm.. Rev., vol. 31, no. 4, pp. 161-172, 2001.

[4] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," Proc. IFIP/ACM Int'l Conf. Distributed Systems Platforms (Middleware), pp. 329-350, Nov. 2001.

[5] G. Networks, "Groove Networks," http://www.groove.net/ products/workspace/securitypdf.gtml, 2009.

[6] R.L. Rivest and B. Lampson, "SDSI: A Simple Distributed SecurityInfrastructure," Proc. Crypto '96, pp. 104-109, Aug. 1996.

[7] N. Li and J.C. Mitchell, "RT: A Role-Based Trust-Management Framework," Proc. Third DARPA Information Survivability Conf. andExposition (DISCEX III), Apr. 2003.

[8] D. Ferraiolo and R. Kuhn, "Role-Based Access Controls," Proc. 15th Nat'l Computer Security Conf., May 1992.

[9] D. Chaum, "Blind Signatures for Untraceable Payments," Proc. Advances in Cryptology (Crypto '82), 1983.

[10] L. Zhou, F. Schneider, and R. Renesse, "COCA: A Secure Distributed Online Certification Authority," ACM Trans. Computer Systems, vol. 20, no. 4, pp. 329-368, Nov. 2002.

[11] M. Chen and J.P. Singh, "Computing and Using Reputations for Internet ratings," Proc. Third ACM Conf. Electronic Commerce, pp. 154-162, 2001.

[12] P. Resnick, R. Zeckhauser, and E. Friedman, "Reputation Systems," Comm. ACM, vol. 43, pp. 45-48, Dec. 2000.

[13] E. Friedman and P. Resnick, "The Social Cost of Cheap Pseudonyms," J. Economics and Management Strategy, vol. 10, no. 2, pp. 173-199, 2001.

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

[15] A. Abdul-Rahman and S. Hailes, "Supporting Trust in Virtual

Communities," Proc. Hawaii Int'l Conf. System Sciences, Jan. 2000.

[16] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," Proc. 10th Int'l Conf. Information and KnowledgeManagement (CIKM '01), pp. 310-317, Nov. 2001.

[17] A.I. Schein, A. Popescul, L.H. Ungar, and D.M. Pennock, "Methods and Metrics for Cold-Start Recommendations," Proc. 25th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 253-260, 2002.

