ISSN:0975-9646

Interfirewall Optimization across Administrative Domains for Enabling Privacy Preserving and Security

Aruna Devi.R[#1], PON Arivanandham[*2]

[#1]*PG Student*
Dhanalakshmi Srinivasan college of Engineering and Technology
Mamallapuram, Chennai

[*2]*Associate professor*
Dhanalakshmi Srinivasan college of Engineering and Technology
Mamallapuram, Chennai

***Abstract*— Enabling security over internet, firewalls play a major role. It checks all incoming or outgoing packet to decide whether to accept or discard the packet based on its policy. Firewall optimization focuses on either intra-firewall or inter-firewall optimization within one administrative domain where the privacy of firewall policies is not a concern. Explore Interfirewall optimization across administrative domains for the first time. The key technical challenge is that firewall policies cannot be shared across domains because a firewall policy contains confidential information and even potential security holes, which can be exploited by attackers. Using Interfirewall redundant rule which overcome the prior problem and enable the Interfirewall optimization across administrative domains. Also propose the first cross-domain cooperative firewall (CDCF) policy optimization protocol. The optimization process involves cooperative computation between the two firewalls without any party disclosing its policy to the other. We implemented our protocol in Java and conducted extensive evaluation.**

***Keywords*— Interfirewall optimization, security, CDCF**

I.INTRODUCTION

A. Background and Motivation

A firewall is defined as any device (or software) used to filter or control the flow of traffic. Firewalls are typically implemented on the network perimeter and function by defining trusted and untrusted zone. Most firewalls will permit traffic from the trusted zone to the untrusted zone, without any explicit configuration. However, traffic from the
Untrusted zone to the trusted zone must be explicitly permitted. Thus, any traffic that is not explicitly permitted from the untrusted to trust zone will be implicitly denied (by default on most firewall systems)**.** The basic purpose of a firewall is to keep uninvited guests from browsing your network. A firewall can be a hardware device or a software application and generally is placed at the perimeter of the network to act as the gatekeeper for all incoming and outgoing traffic. There are basically four mechanisms used by firewalls to restrict traffic. One device or application may use more than one of these in conjunction with each other to provide more in-depth protection. The four mechanisms are packet-filtering, circuit-level gateway, and proxy server and application gateway. Packet Filtering is one of the core services provided by firewalls. Packets can be filtered (permitted or denied) based on a wide range of criteria:

- Source address
- Destination address
- Protocol Type (IP, TCP, UDP, ICMP, ESP, etc.)
- Source Port
- Destination Port

Packet filtering is implemented as a rule-list. The order of the rule-list is a critical consideration. The rule-list is always parsed from top-to-bottom. Thus, more specific rules should always be placed near the top of the rule-list; otherwise they may be negated by a previous, more encompassing rule. Also, an implicit 'deny any' rule usually exists at the bottom of a rule-list, which often can't be removed. Thus, rule-lists that contain only deny statements will prevent all traffic. Normally, message privately over an insecure channel. By an insecure channel, we mean there is an adversary, say Eve (or eavesdropper), who listens everything on this channel. How do we achieve this?

A possible solution: "secret code". A secret code Consists of a key, an algorithm to encrypt (scramble) text and an algorithm to decrypt (Descramble) text. Let us try to formalize this solution. It is clear that we need an algorithm to generate keys (Gen), an encryption algorithm (Enc) and a decryption algorithm (Dec). We also need to decide what is known by everyone (public) and what is kept secret (private). A triplet (Gen, Enc, and Dec) of algorithms, a message space M and a key space K is called a private-key encryption scheme if:

1. The key-generation algorithm: Gen is a randomized algorithm that returns a key k,

Denoted by k ← Gen, such that k ∈K.

2. The encryption algorithm: Enc is an algorithm (potentially randomized) that takes a key k and a plain-text message m ∈ M, and outputs a cipher text c ← Enck (m).

3. The decryption algorithm: Dec is an algorithm that takes a key k and a cipher-text C and outputs a plaintext m.

4. The scheme should satisfy the following property: For all m ∈M and k ∈K,

Pr [Deck (Enck (m)) = m] = 1.

Encryption is a process of coding information which could either be a file or mail message in into cipher text a form unreadable without a decoding key in order to prevent anyone except the intended recipient from reading that data. Decryption is the reverse process of converting encoded data to its original un-encoded form, plaintext. A key in cryptography is a long sequence of bits used by encryption / decryption algorithms. Secret Key Cryptography (SKC): Uses a single key for both encryption and decryption. A firewall configuration is specified as a sequence of rules. Each rule in a firewall configuration is of the form

<Predicate>-><decision>

The <predicate of a rule is a Boolean expression over some packet fields together with the [physical network interface on which a packet arrives. The <decision> of a rule can be accept, or discard, or a combination of these decisions with other options such as a logging option. A packet matches a rule if a firewall configuration overlap if there is at least one packet that can match both rules.

B.Limitation of Prior Work

Previous work on firewall analysis focuses on conflict detection [Hari et al. (2000); Epstein and Muthukrishnan (2001); Moffett and Sloman (1994); Baboescu and Varghese (2002)].The basic idea of firewall conflict detection is to first detect all pairs of rules that conflict, and then the firewall designer manually examines every pair of conflicting rules to see whether the two rules need to be swapped or a new rule needs to be added. Examining each conflict or anomaly is helpful in reducing errors. We approach this goal from two directions:

(1) How to reduce errors when a firewall configuration is being designed.
(2) How to detect errors after a firewall configuration has been designed.

II.CROSS-DOMAIN COOPERATIVE FIREWALL (CDCF)

Firewall works on both intrafirewall and Interfirewall domains. Consider 3 domains and we need to detect Interfirewall redundant rules for these 3 domains. Since firewall policy contains confidential and private information, we need to provide security. Let us consider 3 adjacent firewalls 1, 2 and 3 which belong to different administrative domains D1, D2 and D3. Based on the rule r, Interfirewall redundant rule, check the incoming and outgoing packets among these domains that are F1, F2 and F3 were denoted as firewall policy. A firewall protocol is considered to be a collection or list of rules. In which each rule has a *predicate* over d *fields* $F_1 \ldots F_d$ and a *decision* for the packets that match the predicate. The protocol contains source port, destination port, source IP, destination IP and protocol type and finally reports the action whether it "accept or deny". First convert each firewall F1, F2, F3 into non overlapping rules.

Check the matching set of non overlapping rules nr with resolving set (i.e.) *M(nr)=R(nr)* Here check whether the non overlapping rule *nr* in F2 satisfies the non overlapping discarding rule in F1 and in similar way for F3. And also check for the multiple non overlapping discarding rules. And also need to check Privacy-Preserving Range Comparison. If the rule exists we propose a *cross domain cooperative firewall protocol* to optimize the network. If the rule does not exist, the network performance collapse and discards due to the entry of third party. Thereby privacy and security fails. To overcome this bug we underwent a study of *cross domain cooperative firewall protocol.* The rule optimization from F1 to F2 and F2 to F3 and similar rule optimization is possible in opposite direction F3 to F2 and F2 to F1. Here F1 improves the performance load of F2 and F2 improves the performance load of F3 and vice versa. Since our 3 domains must attain benefit from it and must be explored in a similar mutual manner.

In Cross Domain Cooperative Firewall allows the network to enforce each other across multiple domains (two or more) and regulates the traffic. The protocol is studied here more than two domains (multiple) where privacy have been protected. Optimizing firewall is important for network performance. Same way, Security and privacy are two major concerns in supporting users across administrative domains.

III. KEY CONTRIBUTIONS

We make three key contributions. First, we propose a CDCF protocol for detecting Interfirewall redundancy detection. Second propose an Encryption/Decryption algorithm for security. We implemented our protocol and conducted extensive experiments on both real and synthetic firewall policies. The results on real firewall policies show that our protocol can remove as many as 98% of rules.

In this key contribution we organize the paper as follows. significant new enhancements and various applications are explains about the key distribution, and to its review related work in Section I.Then, proposed work in section II.In section IV the methodology of the concept.Im Section V we give the security analysis of our protocol. In section VI Experimental results is shown. Finally we conclude the firewall optimization shows 98% of rules removal in Section VII. In final we give the reference to successfully done this paper.

IV. METHODOLOGY

The configuration for the proposed system is shown in Figure 1.

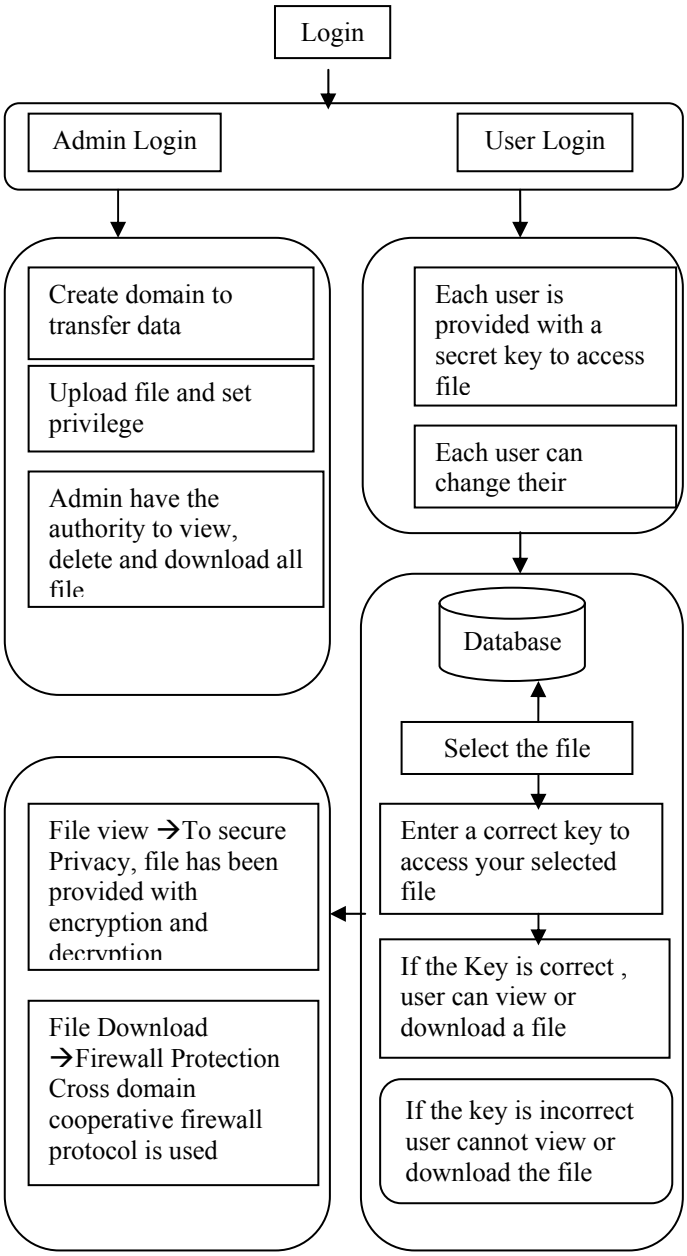

Fig 1: Interfirewall optimization in different Administrative domains

The user enters the application through user login. There are multiple users in our project. Each user must have username and Password. After validating the username and password from the database, the user can able to enter the System and search for specific domain or cross domain search. Also admin have username and password, so that they enter to view the received search information, if admin enter the right username and password.

"An information access system that allows access to all the information on the web that is relevant to a particular domain". A Domain search has much search expertise. Sometimes giving the presence of unreliable information on the Web, this often leads inexperienced users to perform ineffective searches. A cross-domain search provides the ability to manually or automatically access or transfer between two or more differing security domains. Here Interfirewall optimization is possible across one or more administrative domain. The rules in a firewall policy typically follow the first-match semantics, where the decision for a packet is the decision of the first rule that the packet matches in the policy. Each physical interface of a router/firewall is configured with two ACLs: one for filtering outgoing packets and the other one for filtering incoming packets. Security and privacy are major concerns in supporting users across administrative domain.

A firewall consists of rules. It contains Source Port, Destination Port, Source IP, Destination IP, and Protocol and finally the action decides whether to accept or deny the packets. The network should match with the firewall rules. In some cases, the number of rules in a firewall significantly affects its throughput. By increasing the number of rules in the firewall policy it gradually reduces the firewall output. To enable cooperative filtering across administrative domains, one fundamental challenge is to preserve the privacy of different parties. During the transmission of packets from one network to the other with the firewall protection a third party enters into the network to degrade the performance of network. Some malicious activity may be identified by the other party using anomaly detection approaches.

A third party refers to hackers which enters into the network for malicious reasons in order to hack the information and disturbs the network. It is very difficult to surf the internet with these kinds of hackers. Our response is to overcome this attack. Interfirewall helps in preventing hackers or malicious software from gaining access to our system through the internet or network. This Interfirewall also helps to stop our system from sending malicious software to other computers or networks. Here our Interfirewall acts as a barrier between the system and network.

Due to the interference of hackers the firewall consists of a redundant/irrelevant set of rules. In order to remove these unwanted traffic we proposed a method "Cross Domain Cooperative Firewall". This method regulates the traffic and enforces the network firewall rules and yet preserves the privacy and security of all parties involved. The key ingredients in CDCF are the distribution of firewall primitives across network domains, and the enabling technique of efficient method.

A Cross-Domain Cooperative Firewall that allows two networks to collaboratively enforce each other's firewall rules in an oblivious manner.CDCF is used for to remove overall redundancy rules. It is very effective compared to other.CDCF is used to enable the security and filtering the rules without exposing the shared messages.CDCF is used for rule matching if the rule is matched means it allows the

packets, rule can't matched means it discard the packet. It removes the rules in efficient manner. It mainly used for to find location based service.

V.SECURITY ANALYSIS

Public-key cryptography, also known as asymmetric cryptography, refers to a cryptographic algorithm which requires two separate keys one of which is *secret* (or *private*) and one of which is *public*. Although different, the two parts of this key pair are mathematically linked. The public key is used to encrypt plaintext or to verify a digital signature; whereas the private key is used to decrypt cipher text or to create a digital signature. The term "asymmetric" stems from the use of different keys to perform these opposite functions, each the inverse of the other – as contrasted with conventional ("symmetric") cryptography which relies on the same key to perform both. Public-key algorithms are based on mathematical problems which currently admit no efficient solution that are inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships. It is computationally easy for a user to generate his or her public and private key-pair and to use them for encryption and decryption. The strength lies in the fact that it is "impossible" (computationally infeasible) for a properly generated private key to be determined from its corresponding public key.

Thus the public key may be published without compromising security, whereas the private key must not be revealed to anyone not authorized to read messages or perform digital signatures. Public key algorithms, unlike symmetric key algorithms, do *not* require a secure initial exchange of one (or more) secret keys between the parties. Message authentication involves processing a message with a private key to produce a digital signature. Thereafter anyone can verify this signature by processing the signature value with the signer's corresponding public key and comparing that result with the message. Success confirms the message is unmodified since it was signed, and – presuming the signer's private key has remained secret to the signer – that the signer, and no one else, intentionally performed the signature operation. In practice, typically only a hash or digest of the message, and not the message itself, is encrypted as the signature.

The distinguishing technique used in public-key cryptography is the use of asymmetric key algorithms, where the key used to encrypt a message is not the same as the key used to decrypt it. Each user has a pair of cryptographic keys – a public encryption key and a private decryption key. Similarly, a key pair used for digital signatures consists of a private signing key and a public verification key. The public key is widely distributed, while the private key is known only to its proprietor. The keys are related mathematically, but the parameters are chosen so that calculating the private key from the public key is either impossible or prohibitively expensive. We are using public key for privacy preserving and security.

Firewall Decision Diagram

A field F_i is a variable, whose value is taken from a predefined interval of nonnegative integers, called the domain of F_i and denoted by D (F_i). A packet over the fields $F_0 . . . F_{n-1}$ is an n-tuple ($p_0, . . . , p_{n-1}$) where each p_i is taken from the domain D (F_i) of the corresponding field F_i. A firewall decision diagram f over the fields F_0, F_{n-1} is an acyclic and directed graph that satisfies the following conditions: (1) F has exactly one node that has no incoming branches, calls the "root of f", and has two or more nodes that have no outgoing branches, called the "terminal nodes of f". (2) Each non-terminal node ν in f is labeled with a field, denoted by F (ν), taken from the set of fields F_0. F_{n-1}. Each terminal node ν in f is labeled with a decision that is either accept or "a" or discard or "d".(3) A directed path from the root node to a terminal node in f is called a "decision path". No two nodes on a decision path in f have the same label.(4) Each branch e that is an outgoing branch of a node ν in f is labeled with an integer set I (e), where I (e) is a subset of the domain of field F (ν).5) Let ν be any terminal node in f. The set E (ν) of all outgoing branches of node ν satisfies the following two conditions:

(a) Consistency: For any distinct e_i and e_j in E (ν), $I(e_i) \cap I(e_j) = \emptyset$.

(b)Completeness: $\cup_{e \in E(v)} I(e) = D(F(v))$, where Ø is the empty set and D(F(ν)) is the domain of the field F(ν) .

VI.EXPERIMENTAL RESULTS

We evaluate the effectiveness of our protocol on real firewalls and evaluate the efficiency of our protocol on both real and synthetic firewalls. We implemented our protocol using java. Our experiments were carried out on a PC running Windows XP with 20GB of memory.Firewall converts the packets into set of rules. Set of rules nothing but source IP, Destination IP, source port, destination port, prototype type. Firewalls check each incoming and outgoing packets with the rule sets. If there is presence of redundant rules means the third party entry into the network to collapse the network performance so there is traffic in network, the arrival of third party infers that privacy and security fails to block the third party we implement the CDCF protocol. It addressed and modified the firewall optimization in network to overcome traffic and redundancy set and to provide security and privacy.

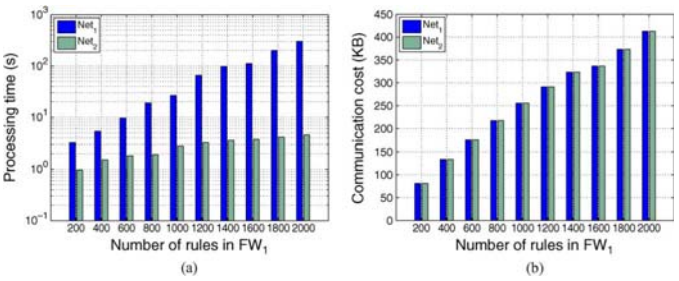

Fig. 2 Processing on synthetic firewalls

The processing time of N1 is less than 1 seconds and the processing time of N2 is less than 1 second. When processing F1 in those real firewall groups, the processing time of Net_1 is less than 2 seconds and the processing time of N_2 are less than 5 seconds. The comparison time of two firewalls is less than 0.01 seconds. The total processing time of two parties is less than 15 seconds, which demonstrates the efficiency of our protocol.

Our protocol is efficient for the communication cost between two parties. When processing firewall F1 in the communication cost from N1 to N2 and that from N2 to N1 are less than 60 KB. Note that the communication cost from N1 to N2 and that from N2 to N1 are the same because N1 and N2 encrypt the same number of values and the encrypted values have the same length, *i.e.,* 1024 bits in our experiments. When processing F2 in those real firewall groups, the communication cost from N2 to N1 is less than 100KB. The total communication cost between two parties is less than 150KB, which can be sent through the current network (*e.g.*, DSL network) around 8 seconds.

Efficiency on Synthetic Firewall Policies:

For the synthetic firewalls, Fig. 2 show the average processing time and communication cost of two parties N1 and N2 for processing F1 and F2, respectively.

Our protocol is efficient for processing and comparing two synthetic firewalls. When processing the synthetic firewalls as F_1, the processing time of N1 is less than 150 seconds and the processing time of N2 is less than 1 seconds. When processing the synthetic firewalls as F_2, the processing time of N1 is less than 150 seconds and the processing time of N2 is less than 5 seconds. The comparison time of two synthetic firewalls is less than 1 seconds.

Our protocol is efficient for the communication cost between two synthetic firewalls. When processing the synthetic firewalls as F_1, the communication cost from N1 to N2 and that from N2 to N1 grow linearly with the number of rules in $F W_1$, and both costs are less than 150 KB. Similarly, when processing synthetic firewalls as F 2, the communication cost from N2 to N1 grows linearly with the number of rules in $F W_2$, and the communication cost from N2 to N1 is less than 100 KB.

VII. CONCLUSION

Firewalls are designed to provide access control. We proposed the method Cross Domain cooperative firewall across different administrative domains by using key management, in order to enable a privacy preserving and security. By using this method the security will be increased and controlled and also we can able to provide privacy and security. We need to check Privacy-Preserving Range Comparison. If the rule exists we propose a cross domain cooperative firewall protocol to optimize the network. If the rule does not exist, the network performance collapse and discards due to the entry of third party. Thereby privacy and security fails. To overcome this bug we underwent a study of cross domain cooperative firewall protocol. We implemented our protocol in java and conducted extensive evaluation. The results on real firewall policies show that our protocol can avoid 98% of rules in a firewall. Future work is in order to increase the system accuracy by extending the current protocol. To find out the maximum speed of the packet to be reached. Extending the process used for proxy server. Using VGuard framework for sending and receiving the packets is very faster.

REFERENCES

[1] Firewall throughput test, www.hipac.org/performance tests/results.html.
[2] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *ACM SIGMOD*, pages 86–97, 2003.
[3] A. X. Liu, E. Torng, and C. Meiners. Firewall compressor: An algorithm for minimizing firewall policies. In *IEEE INFOCOM*, 2008.
[4] C. R. Meiners, A. X. Liu, and E. Torng. Topological transformation approaches to optimizing tcam-based packet processing systems. In *ACM SIGMETRICS*, pages 73–84, 2009.
[5] A. X. Liu and M. G. Gouda. Complete redundancy removal for packet classifiers in tcams. *IEEE TPDS*, in press.
[6] A. X. Liu and F. Chen. Collaborative enforcement of firewall policies in virtual private networks. In *ACM PODC*, pages 95–104, 2008.
[7] E. Al-Shaer and H. Hamed. "Firewall Policy Advisor for Anomaly Detection and Rule Editing." IEEE/IFIP Integrated Management Conference (IM'2003), March 2003.
[8] E. Al-Shaer and H. Hamed. "Design and Implementation of Firewall Policy Advisor Tools." DePaul CTI Technical Report, CTI-TR-02-006, August 2002.
[9] Y. Bartal., A. Mayer, K. Nissim and A. Wool. "Firmato: A Novel Firewall Management Toolkit." Proceedings of 1999 IEEE Symposium on Security and Privacy, May 1999
[10] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda and T. Przygienda. "QoS routing mechanisms and OSPF extensions," *RFC 2676*, http://www.ietf.org/rfc/rfc2676.txt, August 1999.
[11] E. Al-Shaer and H. Hamed. Firewall policy advisor for anomaly detection and rule editing. In Proc. IEEE/IFIP Integrated Management Conference (IM'2003), March 2003.
[12] E. Al-Shaer and H. Hamed. Discovery of policy anomalies in distributed firewalls. In Proc. IEEE Infocomm, Hong Kong, Mar 2004.
[13] W. R. Cheswick, S. M. Bellovin, and A. D. Rubin. Firewalls and Internet Security: Repelling the Wily Hacker. Addison- Wesley, 2003.
[14] M. G. Gouda and X.-Y. A. Liu. Firewall design: consistency, completeness and compactness. In Proc. ICDCS 24, Mar 2004.
[15] A. Rubin, D. Geer, and M. Ranum, *Web Security Sourcebook*, Wiley Computer Publishing, 1997.
[16] A. Mayer, A. Wool, and E. Ziskind, "Fang: A Firewall Analysis Engine," *Proc. IEEE Symp. Security and Privacy* (S&P 2000), IEEE Press, 2000, pp. 177-187.
[17] D. A. Applegate, G. Calinescu, D. S. Johnson, H. Karloff, K. Ligett, and J. Wang. Compressing rectilinear pictures and minimizing access control lists. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, January 2007.
[18] M. G. Gouda and A. X. Liu. Structured firewall design. Computer Networks Journal (Elsevier), 51(4):1106–1120, March 2007.
[19] Y. Bartal, A.J. Mayer, K. Nissim, and A. Wool, "Firmato: A Novel Firewall Management Toolkit," Proc. IEEE Symp. Security and Privacy (S&P '99), pp. 17-31, 1999.
[20] H. Anderson and G. Hagelin, "Computer Controlled Interlocking System," Ericsson Rev., vol. 2, 1981.
[21] J. Lee, J. Jeon, and K. Yoo, "A security scheme for protecting security policies in firewall," *SIGOPS Operating System Review*, vol. 38, no. 2, pp. 69–72, 2004.