Advances in Computer Science and its Applications (ISSN 2166-2924)
Vol. 1, No. 2, June 2012


www.worldsciencepublisher.org

# Nymble: Intercepts Misbehaving Users

**P.Lavanya[1], Manoj Kumar Tyagi[2], K.Poornima[3], N.Mahesh[4]**

[134]Electronics and computers Department, Koneru Lakshmaiah University, Vaddeswaram, Greenfields, Guntur, India

[2]Assoc.Proff, ECM Dept, Koneru Lakshmaiah University, Vaddeswaram, Greenfields, Guntur, India

Email: [1]lavanyapothineni@yahoo.co.in, [2]manojkumar@kluniversity.in, [3]poornima.kollipara@gmail.com, [4]maheshdonofecm@gmail.com

**Abstract**—Every user who wants to connect to a server has to provide his ID details where as some of the users connect through anonymizing networks such as Tor allow users to access Internet services privately by using a series of routers to hide the client's IP address from the server. The success of such networks, however, has been limited by users employing this anonymity for abusive purposes such as defacing popular websites. Website administrators routinely rely on IP-address blocking for disabling access to misbehaving users, but blocking IP addresses is not practical if the abuser routes through an anonymizing network. As a result, administrators block all known exit nodes of anonymizing networks, denying anonymous access to misbehaving and behaving users alike. To address this problem, we present Nymble, a system in which servers can "blacklist" misbehaving users, thereby blocking users without compromising their anonymity. Our system is thus agnostic to different servers' definitions of misbehavior — servers can blacklist users for whatever reason, and the privacy of blacklisted users is maintained.

**Keywords** – Nymble; Blacklist

## I. INTRODUCTION

Nymble is a system which provides security from anonymizing networks. Firstly, the existing system provides internet services to the users irrespective of their behavior. It causes severe threats to the system. There is a possibility of the users to access the server for abusive purposes. It does not provide security and has a chance of misbehaving with the server information. To overcome the problem of misbehavior, nymble has been proposed.

Nymble is a security providing system in which there is a chance of blocking the misbehaving users with the server. It also helps in enhancing the security to the data stored in the server. It provides the features like backward unlinkability, blacklisting, fast access etc. It provides all the following properties: anonymous authentication, backward unlinkability, subjective blacklisting, fast authentication speeds, rate-limited anonymous connections, revocation auditability (where users can verify whether they have been blacklisted), and also addresses the Sybil attack to make its deployment practical. In Nymble, users acquire an ordered collection of nymbles, a special type of pseudonym, to connect to Websites. Without additional information, these nymbles are computationally hard to link and hence, using the stream of nymbles simulates anonymous access to services. Web sites, however, can blacklist users by obtaining a seed for a particular nymble, allowing them to link future nymbles from the same user—those used before the complaint remains unlinkable. Servers can therefore blacklist anonymous users without knowledge of their IP addresses while allowing behaving users to connect anonymously. Our system ensures that users are aware of their blacklist status before they present a nymble, and disconnect immediately if they are blacklisted. Although our work applies to anonymizing networks in general, we consider Tor for purposes of exposition. In fact, any number of anonymizing networks can rely on the same Nymble system, blacklisting anonymous users regardless of their anonymizing network(s) of choice.

## II. SOFTWARES USED

### A. SOFTWARE

- Front End : Java, RMI, JFC (Swing)
- Server : apache-tomcat-6.0.18(Web Server)
- Backend : MS-Access
- Tools Used : Eclipse 3.3
- Operating System: Windows XP/7

### A. The Java Platform

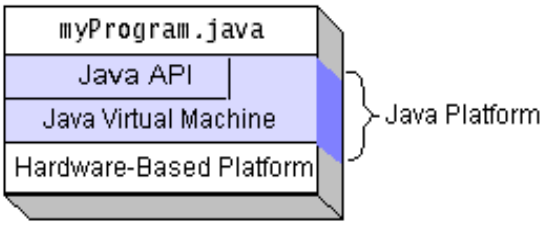
### B. Java Platform

1) A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

2) The Java platform has two components:
   - The *Java Virtual Machine* (Java VM)
   - The *Java Application Programming Interface* (Java API)

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The following figure depicts a program that's running on the Java platform. As the figure shows, the

Java API and the virtual machine insulate the program from the hardware.




Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

## III. OVERVIEW

The project illustrated in this paper is entirely based on the idea of designing a secure system with high-level overview of the Nymble system, and defer the entire protocol description and security analysis to subsequent sections.

### *A. Resource Based Blocking*

To limit the number of identities a user can obtain the Nymble system binds nymbles to resources that are sufficiently difficult to obtain in great numbers. For example, we have used IP addresses as the resource in our implementation, but our scheme generalizes to other resources such as email addresses, identity certificates, and trusted hardware. We do not claim to solve the Sybil attack. This problem is faced by any credential system and we suggest some promising approaches based on resource-based blocking since we aim to create a real-world deployment.

### *B. Blacklisting Anonymous Users*

We provide a means by which servers can blacklist users of an anonymizing network while maintaining their privacy. Practical performance. Our protocol makes use of inexpensive symmetric cryptographic operations to significantly outperform the alternatives.

### *C. Open-Source Implementation.*

With the goal of contributing a workable system, we have built an open-source implementation of Nymble, which is available. We provide performance statistics to show that our system is indeed practical. Some of the authors of this paper have published two anonymous authentication schemes, BLAC and PEREA which eliminate the need for a trusted third party for revoking users. While BLAC and PEREA provide better privacy by eliminating the TTP, Nymble provides authentication rates that are several orders of magnitude faster than BLAC and PEREA. Nymble thus represents a practical solution for blocking misbehaving users of anonymizing networks

## IV. MODULES IMPLEMENTED

- Nymble Manager
- Pseudonym Manager
- Blacklisting a user
- Nymble-authenticated connection

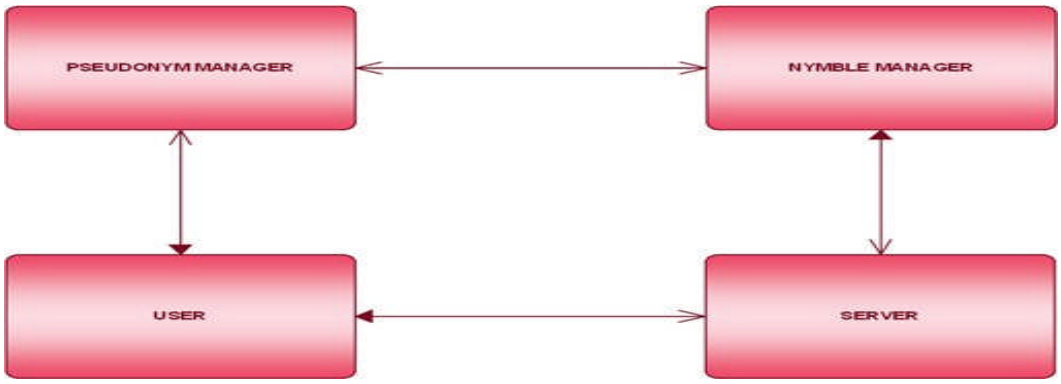



Fig 1 overall system flow

### *A. The Pseudonym Manager*

The user must first contact the Pseudonym Manager (PM) and demonstrate control over a resource; for IP-address blocking, the user must connect to the PM directly (i.e., not through a known anonymizing network), We assume the PM has knowledge about Tor routers, for example, and can ensure that users are communicating with it directly.6 Pseudonyms are deterministically chosen based on the controlled resource, ensuring that the same pseudonym is always issued for the same resource. Note that the user does not disclose to what server he or she intends to connect to, and the PM's duties are limited to mapping IP addresses (or other resources) to pseudonyms. As we will explain, the user contacts the PM only once per linkability window (e.g., once a day).

*B. The Nymble Manager*

After obtaining a pseudonym from the PM, the user connects to the Nymble Manager (NM) through the anonymizing network, and requests nymbles for access to a particular server (such as Wikipedia). A user's requests to the NM are therefore pseudonymous, and nymbles are generated using the user's pseudonym and the server's identity. These nymbles are thus specific to a particular user-server pair. Nevertheless, as long as the PM and the NM do not collude, the Nymble system cannot identify which user is connecting to what server; the NM knows only the pseudonym-server pair, and the PM knows only the user identity-pseudonym pair. To provide the requisite cryptographic protection and security properties, the NM encapsulates nymbles within nymble tickets. Servers wrap seeds into linking tokens, and therefore, we will speak of linking tokens being used to link future nymble tickets. The importance of these constructs will become apparent as we proceed.

*C. Blacklisting a User*

If a user misbehaves, the server may link any future connection from this user within the current linkability window (e.g., the same day). Consider an example: A user connects and misbehaves at a server during time period t_ within linkability window w_. The server later detects this misbehavior and complains to the NM in time period tc (t_ < tc _ tL) of the same linkability window w_. As part of the complaint, the server presents the nymble ticket of the misbehaving user and obtains the corresponding seed from the NM. The server is then able to link future connections by the user in time periods tc; tc þ 1; . . . ; tL of the same linkability window w_ to the complaint. Therefore, once the server has complained about a user, that user is blacklisted for the rest of the day, for example (the linkability window). Note that the user's connections in t1; t2; . . . ; t_; t_ þ 1; . . . ; tc remain unlinkable (i.e., including those since the misbehavior and until the time of complaint). Even though misbehaving users can be blocked from making connections in the future, the users' past connections remain unlinkable, thus providing backward unlinkability and subjective blacklisting.

## OVERVIEW OF NYMBLE

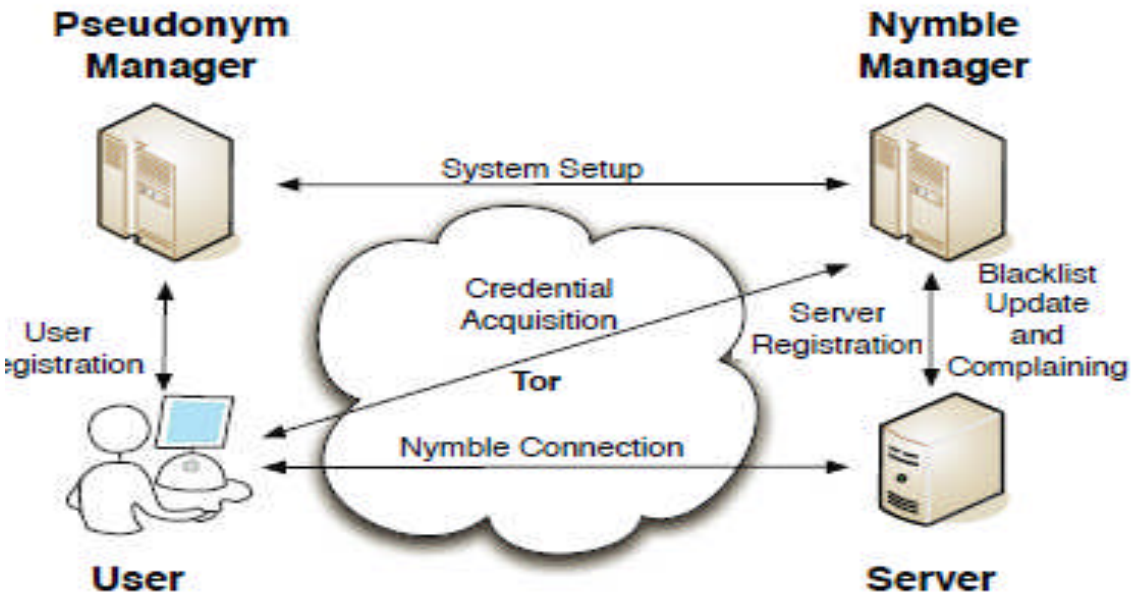



Fig 2 The Nymble system architecture showing the various modes of interaction . Note that users interact with the NM and servers through the anonymizing network

## V. INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

*A. Objectives*

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

*B. Output Design*

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

## VI. OUR NYMBLE CONSTRUCTION AND FEATURES

*A. System setup*

To set up the Nymble system, the NM and the PM interact as follows.

1. The NM executes NMInitState() and initializes its state nmState to the algorithm's output.

2. The NM extracts macKeyNP from nmState and sends it to the PM over a type-Auth channel. macKeyNP is a shared secret between the NM and the PM, so that the NM can verify the authenticity of pseudonyms issued by the PM.

3. The PM generates nymKeyP by running Mac.KeyGen() and initializes its state pmState to the pair (nymKeyP , macKeyNP).

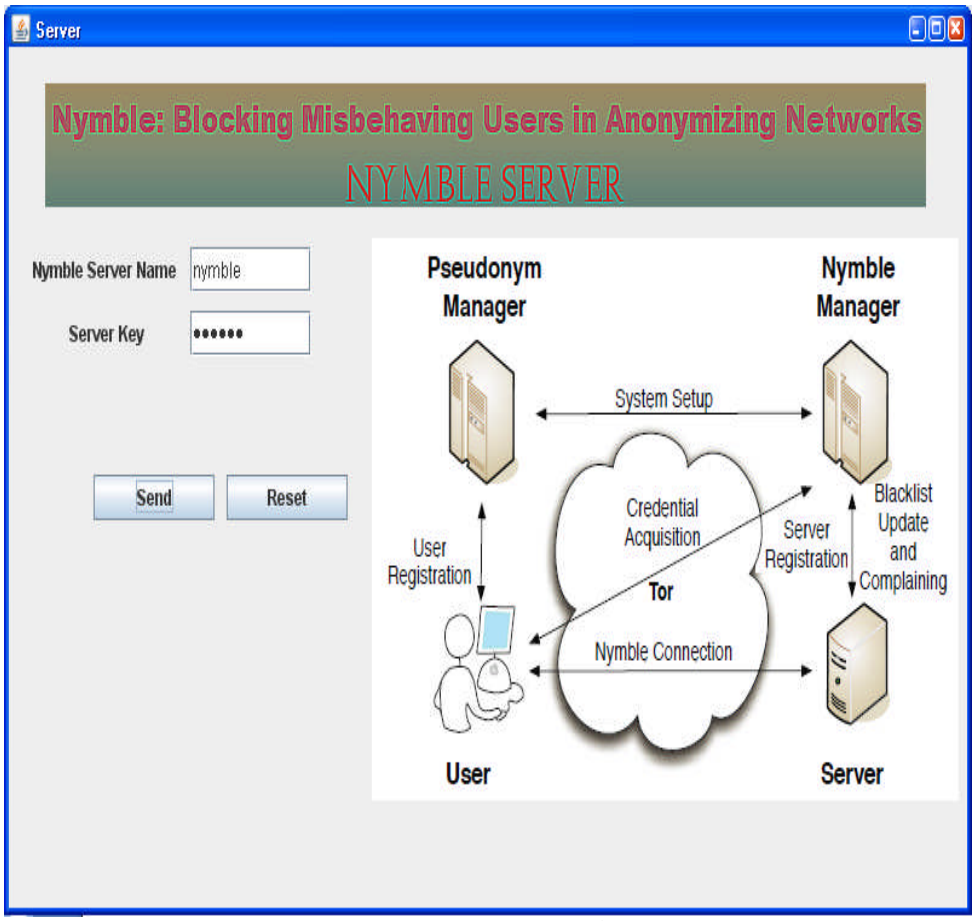



Fig 3 Nymble server login form

4. The NM publishes verKeyN in nmState in a way that the users in Nymble can obtain it and verify its integrity at any time (e.g., during registration).

*Blacklistability* assures that any honest server can indeed block misbehaving users. Specifically, if an honest server complains about a user that misbehaved in the current linkability window, the complaint will be successful and the user will not be able to "nymble-connect," i.e., establish a Nymble-authenticated connection, to the server successfully in subsequent time periods (following the time of complaint) of that linkability window.

*Rate-limiting* assures any honest server that no user can successfully nymble-connect to it more than once within any single time period.

*Anonymity* A user is legitimate according to a server if she has not been blacklisted by the server, and has not exceeded the rate limit of establishing Nymble-connections. Honest servers must be able to differentiate between legitimate and illegitimate users. Anonymity protects the anonymity of honest users, regardless of their legitimacy according to the (possibly corrupt) server; the server cannot learn any more information beyond whether the user behind (an attempt to make) a nymble-connection is legitimate or illegitimate.

*Non-frameability* guarantees that any honest user who is legitimate according to an honest server can nymble-connect to that server. This prevents an attacker from framing a legitimate honest user, e.g., by getting the user blacklisted for someone else's misbehavior.

## VII. SCREENSHOTS

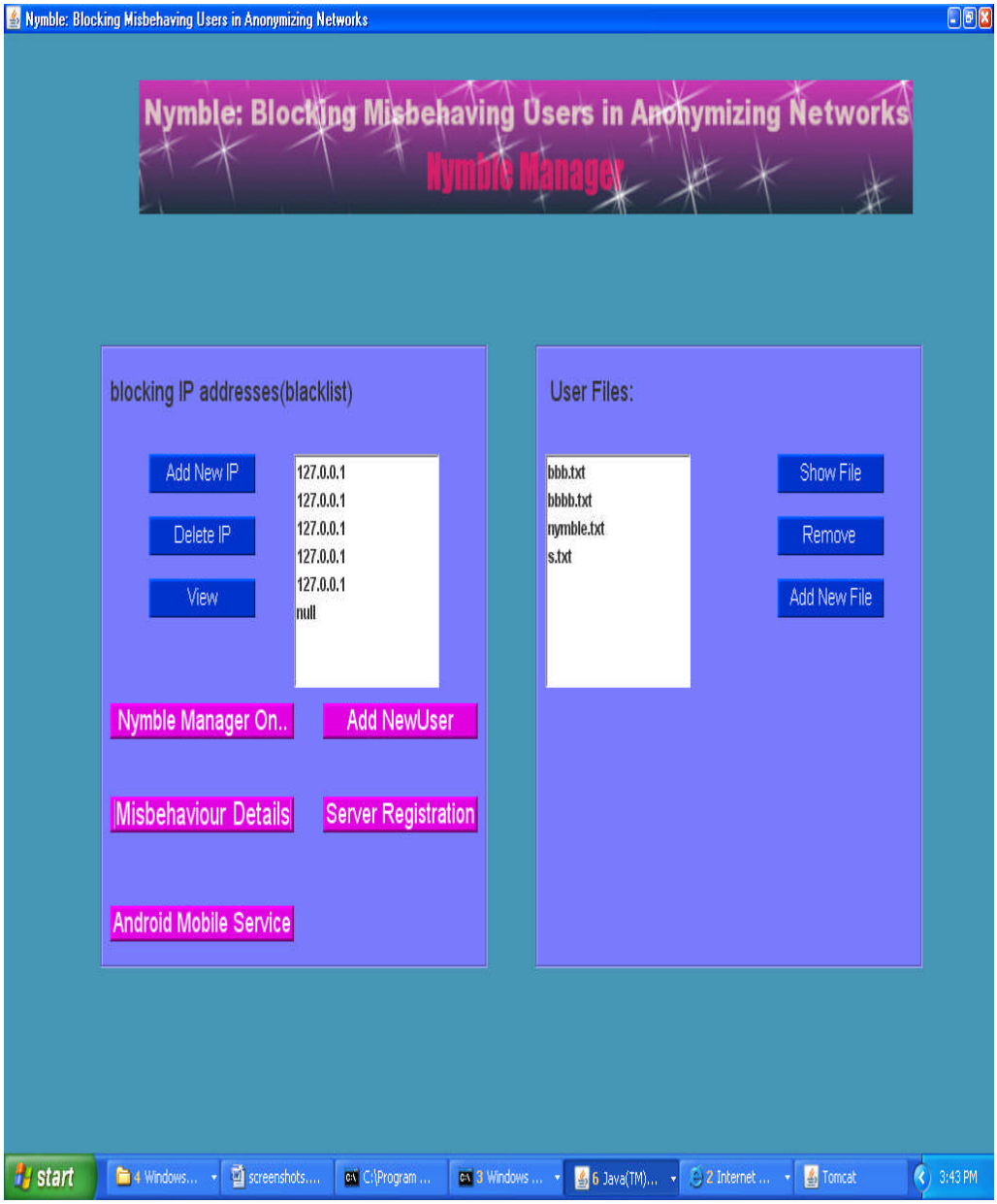



Fig 4 server home page where we can do all the operations

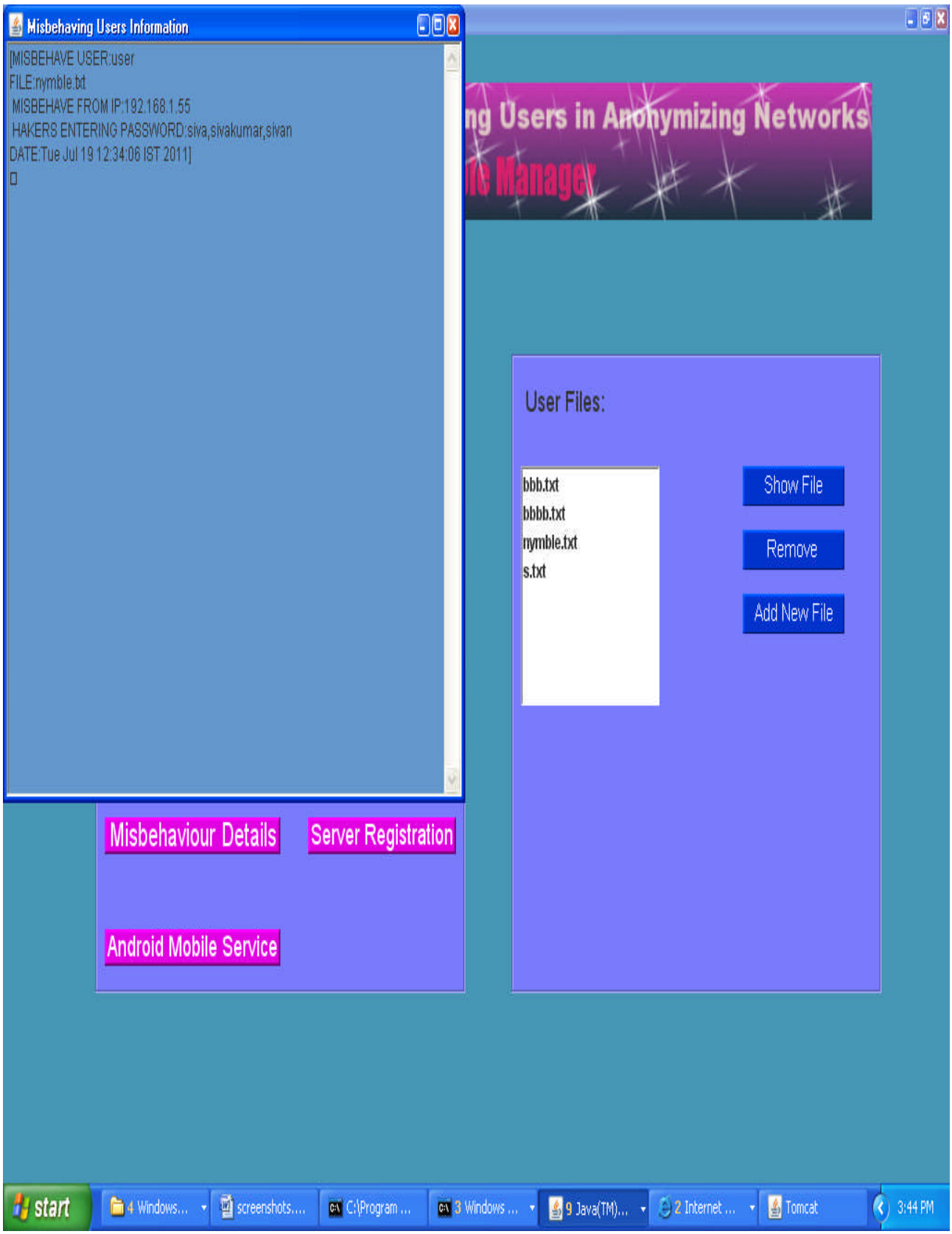



Fig 5 server showing misbehaviour details

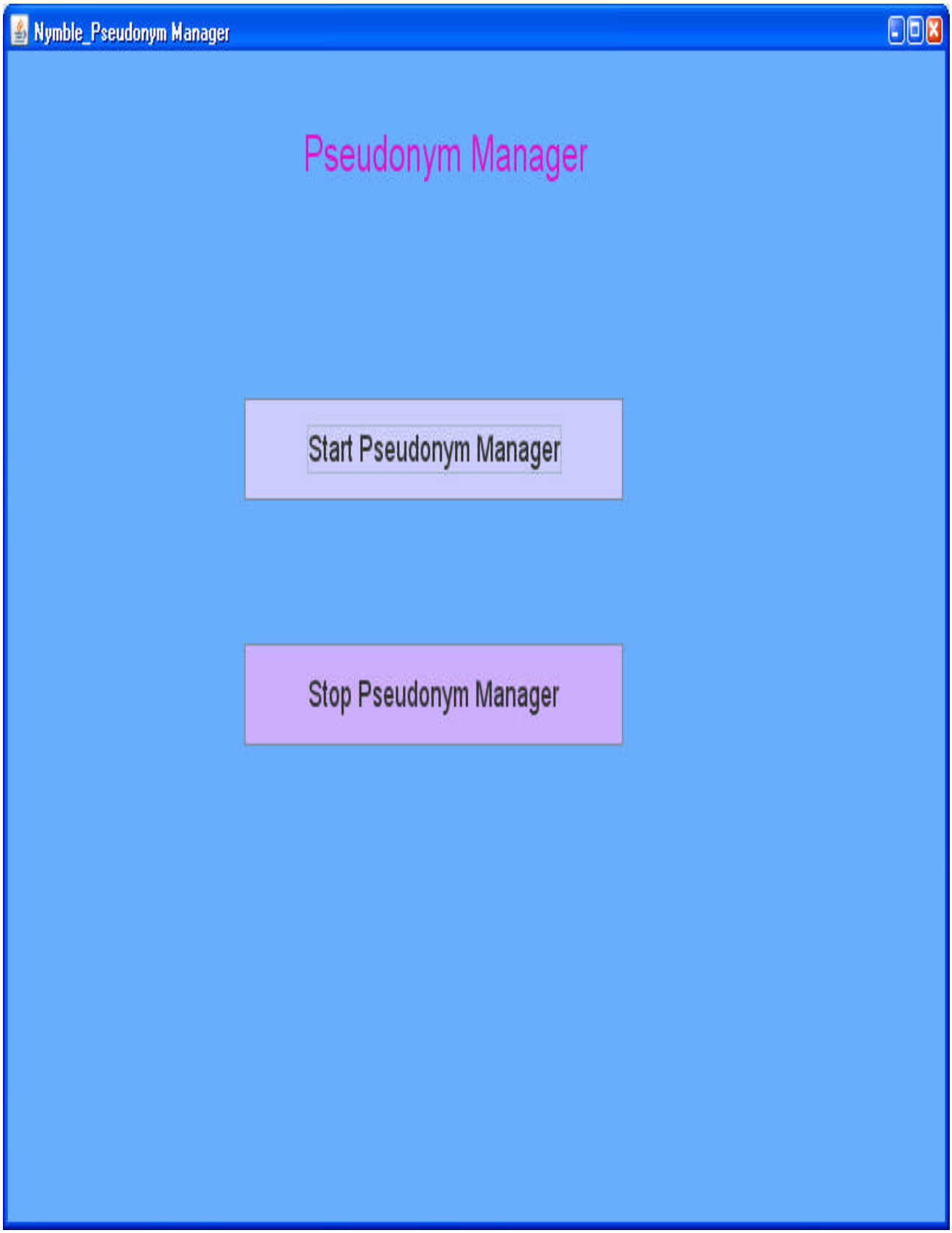



Fig 6 To start pseudonym manager

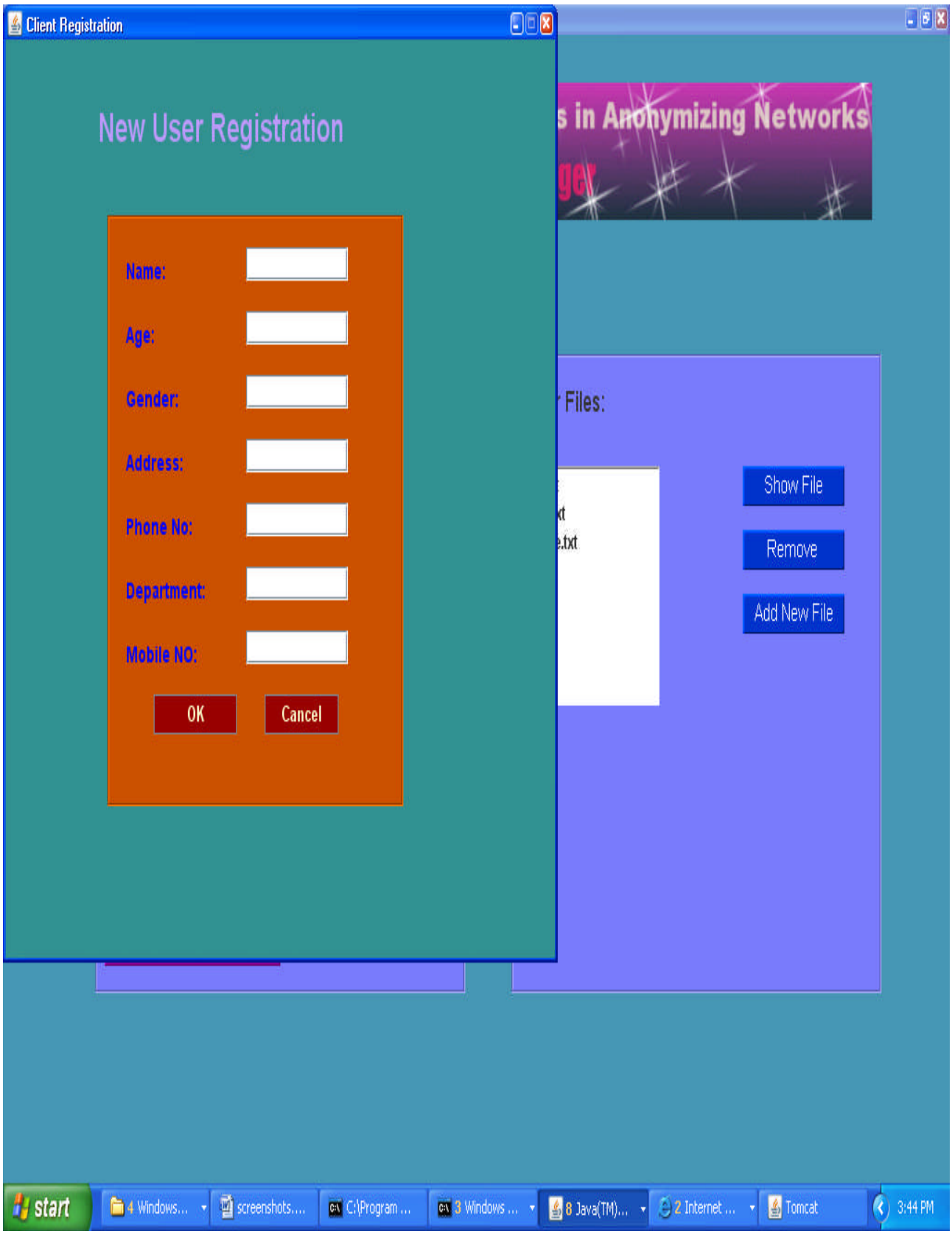



Fig 7 To add new user

## VIII. CONCLUSION

We have proposed and built a comprehensive credential system called Nymble, which can be used to add a layer of accountability to any publicly known anonymizing network. Servers can blacklist misbehaving users while maintaining their privacy, and we show how these properties can be attained in a way that is practical, efficient, and sensitive to the needs of both users and services. We hope that our work will increase the mainstream acceptance of anonymizing networks such as Tor, which has, thus far, been completely blocked by

several services because of users who abuse their anonymity [11-13].

## IX. FUTURE SCOPE

Our nymble project can be extended in wide range and also can be developed on android platform.

## X. ACKNOWLEDGMENTS

We are greatly indebted to our college Koneru Lakshmaiah College of Engineering that has provided a healthy environment to drive us to do this project and thankful to our management for their guidence

## REFERENCES

[1] M. Bellare, H. Shi, and C. Zhang, "Foundations of Group Signatures: The Case of Dynamic Groups," Proc. Cryptographer's Track at RSA Conf. (CT-RSA), Springer, pp. 136-153, 2005
[2] D. Boneh and H. Shacham, "Group Signatures with Verifier-Local Revocation," Proc. ACM Conf. Computer and Comm. Security, pp. 168-177, 2004.
[3] J. Camenisch and A. Lysyanskaya, "An Efficient System for Non-Transferable Anonymous Credentials with Optional Anonymity Revocation," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT), Springer, pp. 93-118, 2001.
[4] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," Proc. Usenix Security Symp., pp. 303- 320, Aug. 2004.
[5] J.E. Holt and K.E. Seamons, "Nym: Practical Pseudonymity for Anonymous Networks," Internet Security Research Lab Technical Report 2006-4, Brigham Young Univ., June 2006.
[6] J. Feigenbaum, A. Johnson, and P.F. Syverson, "A Model of Onion Routing with Provable Anonymity," Proc. Conf. Financial Cryptography, Springer, pp. 57-71, 2007.
[7] P.C. Johnson, A. Kapadia, P.P. Tsang, and S.W. Smith, "Nymble: Anonymous IP-Address Blocking," Proc. Conf. Privacy Enhancing Technologies, Springer, pp. 113-133, 2007.
[8] T. Nakanishi and N. Funabiki, "Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), Springer, pp. 533-548, 2005.
[9] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A Practical and Provably Secure Coalition-Resistant Group Signature, Scheme," Proc. Ann. Int'l Cryptology Conf. (CRYPTO), Springer, pp. 255-270, 2000.
[10] G. Ateniese, D.X. Song, and G. Tsudik, "Quasi-Efficient Revocation in Group Signatures," Proc. Conf. Financial Cryptography, Springer, pp. 183-197, 2002.
[11] Y. Zhang, L. Wu, "A Robust Hybrid Restarted Simulated Annealing Particle Swarm Optimization Technique", Advances in Computer Science and its Applications, vol.1, no.1, pp. 5-8, 2012
[12] Jian Yuan, "A Feedback Linearization based Leader-follower Optimal Formation Control for Autonomous Underwater Vehicles", Advances in Computer Science and its Applications, vol.1, no.1, pp. 45-48, 2012
[13] Iman Sadeghkhani, Elham Hezare, Nima Haratian, "Radial Basis Function based Approach to reduce Shunt Reactor Switching Overvolatges", Advances in Computer Science and its Applications, vol.1, no.1, pp. 49-54, 2012