International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 1, Issue 4 (sept-oct 2013), PP. 27-34

BLACKLISTING MISBEHAVING USERS WHILE MAINTAINING ANONYMITY AND BACKWARD UNLIKABILITY.

V.Spurthi[1], S. Vani Kumari[2]

[1]Computer Science and Engineering Department, GMRIT, Rajam, Srikakulam Dst., India
[2]Asst.Proff, CSE Dept, GMRIT Engineering College, Rajam, Srikakulam Dst., India

Abstract -There exists a plethora of information and communication channels on the web which include a lot of sensitive conversations that people may like to be part of as an anonymous user. Several anonymous networks such as TOR provide this level of anonymity by masking the user's credentials and using one of the many exit nodes to connect to the end user (often a website). However, in case of misbehaviour via such networks, the only way to block such users is to restrict all the exit nodes of anonymous network. This would restrict anonymous access for behaving users as well. In order to resolve this issue, we propose Nymble, a system that supports anonymous blacklisting of misbehaving users. Nymble stays true to the core concepts of anonymous networks such as anonymous usage, backward non-linkability while still providing a mechanism to selectively block misbehaving users. This system is scalable and performs at speeds that is suitable for standard web usage. Our system is thus agnostic to different servers 'definitions of misbehaviour — servers can blacklist users for whatever reason, and the privacy of blacklisted users is maintained.

Keywords –Nymble,Blacklist, information & communication etc

I. INTRODUCTION.

Anonymizing networks work on the concept of onion routing wherein a single message is being sent through various layers of masking. This process ensures that no layer within the path has complete knowledge of the sender or the message and thereby ensure backward unlinkability. Also, as the end point is one of the many exit nodes of the network, this ensures anonymity of the users. However, this creates a major problem for web servers who wish to blacklist misbehaving users. Prior to Nymble, several other approaches were to resolve the issue of block listing mis behaving users while maintaining anonymity, but these systems were marred by severe drawbacks.

The suggested solutions and approaches were:-

(a)Pseudonymous credential systems: Each users were provided a pseudonym using which they could anonymously access a web location and in case of misbehaviour that particular pseudonym is blocked. Though this solution provides proxy-anonymity, it's not truly anonymous since there is a direct one to one relation between the user and pseudonym thereby defeating the purpose of total anonymity.

(b)Anonymous credential systems: In this system, anonymity is provided via group signatures. Essentially a set of users would use a common group signature thereby ensure individual anonymity and the group manager would be notified of the misbehaving user and he in turn would block the user. Though this system serves all the key purposes, the process of contacting the group manager every time to block users was not practically feasible and is a non-scalable solution.

(c)Traceable signatures: In order to improve scalability of group signatures, this particular approach can be used wherein the group manager release a trap door that allows tracing signatures of individual users.

II. SOLUTION: NYMBLE

In order to resolve all the above issues, we suggest a new approach - 'Nymble'. Nymble, through a combination of Nymble server and Pseudonym server ensures that the key elements of anonymous networks still remain while adding the blacklisting ability. Nymble is a system which provides security from anonymizing networks. 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 network can rely on the same Nymble system, blacklisting anonymous users regardless of their anonymizing network(s) of choice.

III. Nymble Architecture:

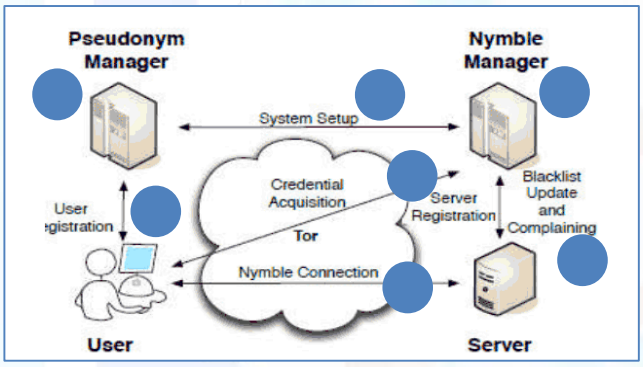

Fig 1:The Nymble system architecture showing the various modes of interaction . Note that users interact with the NM and servers through the anonymizing network

The proposed system for blocking misbehaving users involves some major steps. They are as follows. All users who intend to use anonymizing networks would first register themselves with the pseudonym manager. The pseudonym manager generates a pseudonym for the user. There will be a unique pseudonym-user pair and only the pseudonym manager would have this information. The user collects the pseudonym and contacts the Nymble server. The Nymble server uses a seed (Ex:- A random number) to generate a series of Nymbles. It packages the series of Nymbles into a Nymble ticket and shares it with the user. The Nymble ticket - Pseudonym combination is unique and only the Nymble server is aware of this combination. The user uses this Nymble ticket and contacts the website. Behaving users can thereby ensure complete anonymity and backward unlinkability. In case of a misbehaviour, the website server contacts Nymble server and raises the issue. The Nymble server in such case shares the seed for the corresponding Nymble and the server can block all the subsequent Nymbles that are generated using the seed. Please note the Nymble server only shares the seed and hence, only the future sessions of misbehaving users could be blocked thereby ensuring anonymity. However, once the server obtains the seed, it could backtrack based on this and identify the user. In order to prevent backward linkability, users would be notified on their blacklisting status immediately at the time of sharing the seed with the server. This approach enables the users to be aware of their status and disconnect from the server thereby ensuring backward unlinkability. Also, each Nymble has an associated timeframe and once a server has complained about a user, the blacklist continues for the linkability window. Post expiration of the linkability window, the user can reconnect and continue browsing or communicating anonymously. A system having properties like anonymous authentication, backward unlink ability, subjective blacklisting, and fast authentication speeds, rate limited anonymous connections and revocation auditability is introduced. These properties can be implemented by introducing two trusted third parties namely Pseudonym Manager and Blacklist Manager. The system architecture consists of user, service provider and trusted third parties the blacklist manager and the pseudonym manager. If the user wants to execute web transaction, the user first registers with the Pseudonym Manager and issues user with pseudonym based on the IP address provided by it. The service provider registers with the Blacklist Manager which issues set of unique set of tokens. The user using its pseudonym name access the service provider through the anonymizing network. The service provider transfers the pseudonym to the Blacklist Manager. The Blacklist Manager consists of blacklist table. It has attributes like pseudonym, unique token, and blacklist value. Before the Service Provider give access, it checks with the blacklist table, if the pseudonym is present in the blacklist table, then the user is denied access to SP, else the user can access freely through the network and the Service provider.

IV. Design Issues:

The proposed system should be constructed in such a way that all the entities in the system should be honest. An entity is honest when it's operations are performed according to the system's specification. An honest entity becomes corrupt when it is compromised by an attacker. Once it get compromised then the entity will operate under the full control of the attacker and starts functioning against the system's specification. The proposed system should also satisfy the following security properties. They are

Blacklist ability:

This property assures that any honest server can block misbehaving users. If an honest server complains that a user misbehaved in the current time period, then the complaint will be successful and the user will not be able

to establish a connection to the server successfully for the following time periods.

Rate limiting:

This property assures that any honest server can prevent the user from the successful connection to it, when user attempts to connect to the server more than once within any single time period.

Non-frame ability:

This property assumes that each user has a single unique identity, since it is possible for the user to frame some other identities. So any honest server can provide connection to the server only if it is proved to be an honest user. According to any honest server, a user is honest if he/she has not been blacklisted by the server thus far and has not exceeded the rate limit of establishing connections.

Anonymity:

This property protects the anonymity of honest users such that the server cannot know any information about the user.

V. MODULES IMPLEMENTED:

(i).Server Registration
(ii).User Registration
(iii).Pseudonym Manager
(iv) Nymble Manager
(iiv).Blacklisting a user
(iiiv).Nymble-authenticated connection

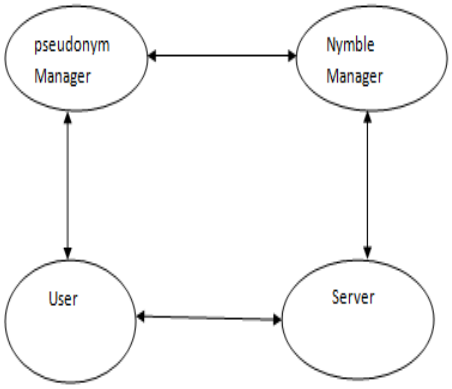

Fig 2 :Overall system flow

Module Description:

Server Registration:

To participate in the Nymble system, a server with identity Sid initiates a type-Auth channel to the NM, and registers with the NM according to the Server Registration protocol below. Each server may register at most once in any likability window. Logins may be used to provide credentials when creating a client connection. Whether or not logins are required depends on the method calls used to start the server or create the connection. For example, you might need logins for pooling. If you do not use logins, you must track and specify the user credentials manually.

User Registration:

A user with identity uid must register with the PM once in each likability window. To do so, the user initiates a type-Basic channel to the PM, followed by the User Registration protocol described below. A login generally requires the user to enter two pieces of information, first a user name and then a password. This information is entered into a login window on a GUI. A user name, also referred to as an account name, is a string (i.e., sequence of characters) that uniquely identifies a user. User names can be the same as or related to the real names of users, or they can be completely arbitrary. A password is likewise a string, but it differs from a user name in that it is intended to be kept a secret that is known only to its use.

***Pseudonym Manager*:**

The user must first contact the Pseudonym Manager and demonstrate control over a resource; for IP-address blocking, the user must connect to the PM directly. We assume the PM has knowledge about Tor routers, and can ensure that users are communicating with it directly. Pseudonyms are deterministically chosen based on the controlled resource, ensuring that the same pseudonyms always issued for the same resource.

Nymble Manager*:*

After obtaining a pseudonym from the PM, the user connects to the Nymble Manager 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.

Blacklist Update:

Servers update their blacklists for the current time period for two purposes. First, as mentioned earlier, the server needs to provide the user with its blacklist (and blacklist certificate) for the current time period during a Nymble connection establishment. Second, the server needs to be able to blacklist the misbehaving users by processing the newly filed complaints (since last update).X-axis refers to the number of entries— complaints in the blacklist update request, tickets in the credential, tokens and seeds in the blacklist update response, and nymbles in the blacklist.

Time:

Tokens generated by the blacklist manager are bound to specific time periods called the likability window. This likability window is again divided small time intervals. Users' access within a time period is tied to single token generated by the Blacklist Manager. The use of different tokens across time periods grants the user anonymity between time periods — smaller time periods provide users with higher rates of anonymous authentication, and likewise longer time periods rate-limit the number of misbehaviours from a particular user before he or she is blocked.

Nymble tickets are bound to specific time periods. As illustrated in Fig.3 , time is divided into linkability windows of duration W, each of which is split into L time periods of duration T (i.e., W ¼L _ T). We will refer to time periods and linkability windows chronologically as t1; t2; . . . ; tL and w1; w2; . . . , respectively. While a user's access within a time period is tied to a single nymble ticket, the use of different nymble tickets across time periods grants the user anonymity between time periods. Smaller time periods provide users with higher rates of anonymous authentication, while longer time periods allow servers to rate-limit the number of misbehaviours from a particular user before he or she is blocked. For example, T could be set to five minutes, and W to one day (and thus, L ¼ 288). The linkability window allows for dynamism since resources such as IP addresses can get reassigned and it is undesirable to blacklist such resources indefinitely, and it ensures forgiveness of misbehaviour after a certain period of time. We assume all entities are time synchronized (for example, with time.nist.gov via the Network Time Protocol (NTP)), and can thus calculate the current linkability window and time period.

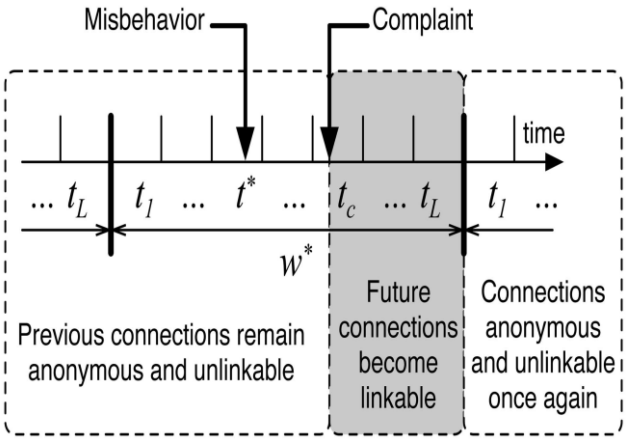

Fig. 3. The life cycle of a misbehaving user. If the server complains in time period tc about a user's connection in t_, the user becomes linkable starting in tc.

VI. DESIGN:

A. Input Design:

The input design is the link between the information system and the user. 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. 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 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.. 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. 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. Select methods for presenting information.

.

Assumptions

1. Pseudonym manager and Nymble manager servers are honest and do not collude to identify the Nymble-Pseudonym-User combination. Such collusion is the only way a user can be tracked.
2.The Pseudonym manager is smart enough to identify users requesting for pseudonyms via anonymizing networks and blocks contact with such users.
3. Nymble server uses an approach to restrict the number of identities an user can own. Unless this is done, user can use successful Sybil attacks to bypass the benefits of current framework and continue misbehaviour without getting blocked.

The above assumptions are practical and feasible and as long as they hold good, Nymble offers complete

anonymiting and the ability to blacklist users in a scalable and fast manner

VII. NYMBLE CONSTRUCTION :

To set up the Nymble system, the NM and the PM interact as follows.
(a). The NM executes NMInitState() and initializes its state nmState to the algorithm's output.
(b). 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.
(c). The PM generates nymKeyP by running Mac.KeyGen() and initializes its state pmState to the pair (nymKeyP , macKeyNP).
(d). 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).

VIII. SCREENSHOTS:

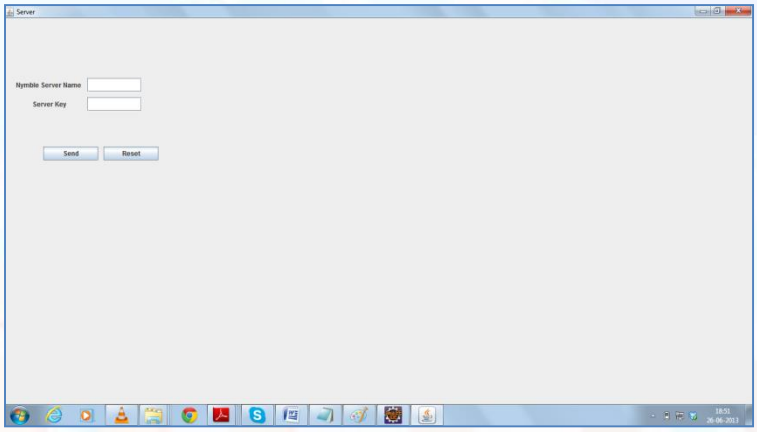

Fig 4: Nymble server login form

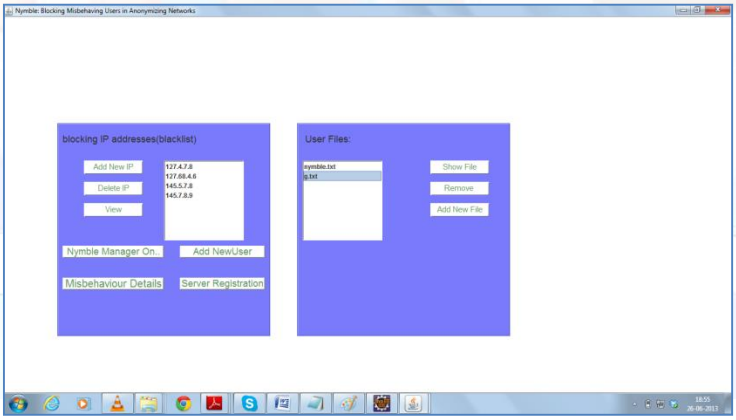

Fig5 : server home page where we can do all the operations

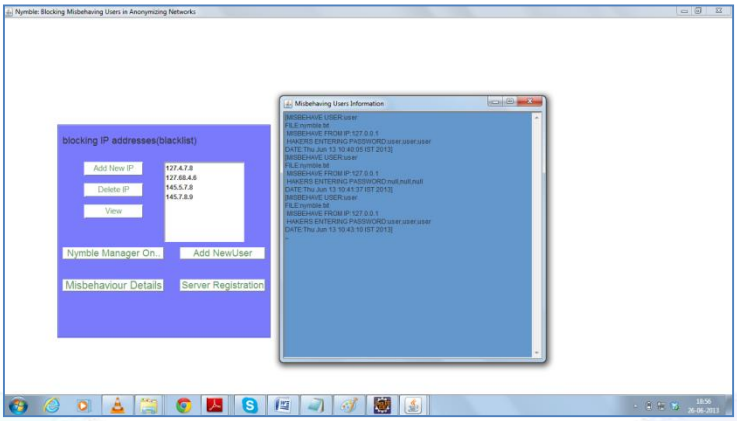

Fig 6:server showing misbehaviour details

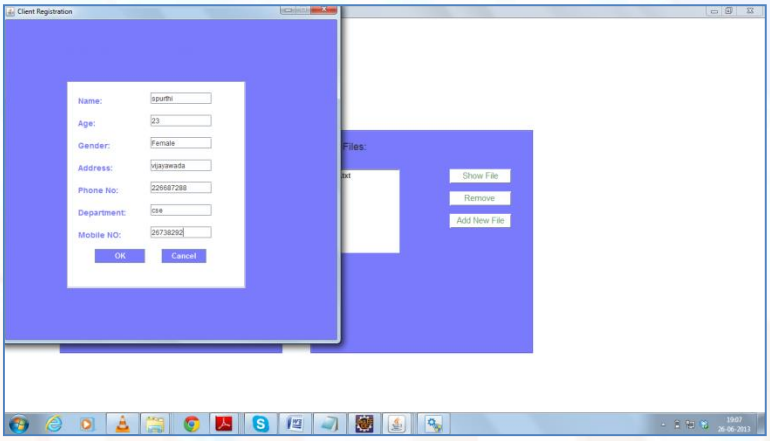

Fig 7: To add new user

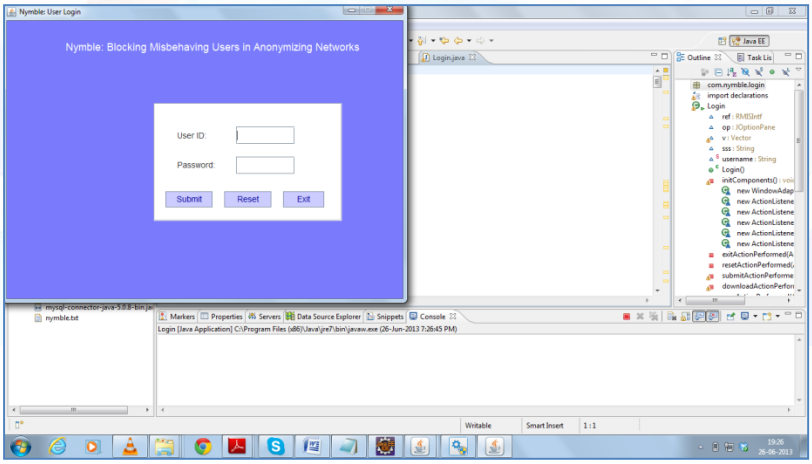

Fig 8:User Login

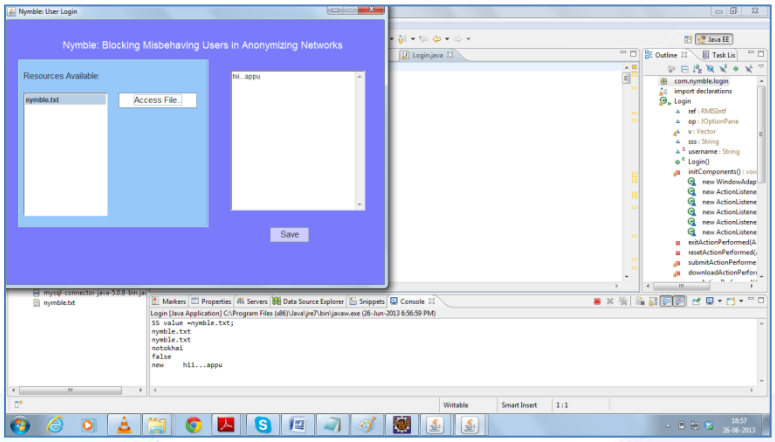

Fig 9: User access file

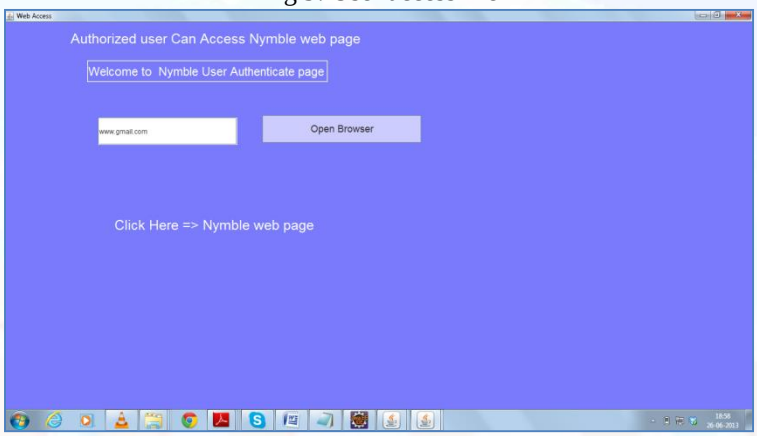

Fig 10:Web Usage through Nymble

IX. CONCLUSION:

A new system is proposed that adds an additional layer of security to the anonymous networks. This system is used to block the misbehaving users in anonymizing networks. It is automatically finds the misbehaving user and blacklists them without affecting their privacy and anonymity. This method is a cryptographic construction that provides anonymous authentication, fast authentication speeds, subjective blacklisting, and backward anonymity and revocation audit ability. This method is practical, effective and efficient to the needs of both users and services. 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.

X. FUTURE SCOPE:

->Platform agnostic implementation of the current framework so that this could be used anywhere.
->Blacklist transferability between collaborating servers to prevent access to repeat offenders.

XI. ACKNOWLEDGMENTS:

We are greatly indebted to our college GMRIT that has provided a healthy environment to drive us to do this project and thankful to our management for their guidance.

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.