Volume 4, Issue 11, November 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Comparative Study of Nymble

Nirav Shah
Rajiv Gandhi Institute of Technology
Mumbai University, Mumbai, India

Saumya Lahera
Shah & Anchor Kutchhi Engineering College
Mumbai University, Mumbai, India

Abstract— In order to allow users to access Internet services privately, anonymizing networks like Tor uses a series of routers to hide the client's IP address from the server. These networks, however, have been marred by users employing this anonymity for abusive purposes such as defacing popular web sites. Usually, web site administrators rely on IP-address blocking in order to disable access to misbehaving users, but it is impractical if the abuser routes through an anonymizing network. In order to avoid this, administrators bar all known exit nodes of the anonymizing network, thereby denying anonymous access to all the users (whether misbehaving or not). To solve this issue, we introduce Nymble, a system where servers blacklist misbehaving users, thereby blocking users without affecting their anonymity. Nymble is thus agnostic to varied definitions of misbehaviour. Servers can block users for any reason, and the privacy of blacklisted users is not affected in any case.

Keywords— anonymous blacklisting, privacy

I. INTRODUCTION

An anonymity network enables users to access the Web while blocking any tracking or tracing of their identity on the Internet [11]. This type of online anonymity moves Internet traffic through a worldwide network of volunteer servers. Anonymity networks prevent traffic analysis and network surveillance - or at least make it more difficult.Anonymity matters because Tor protects you by bouncing your communications around a distributed network of relays run by volunteers all around the world: it prevents somebody watching your Internet connection from learning what sites you visit, and it prevents the sites you visit from learning your physical location.Anonym zing example: Stripping the identity of the sender of an email, from the identification data that accompanies it. The email then becomes an anonymous message. Anonymity software examples are Tor, Freenet, I2P, etc.

Onion routing [13] is a technique for anonymous communication over a computer network. Anonymizing networks such as Tor route traffic through independent nodes in separate administrative domains to hide a client's IP address. Tor protects you by bouncing your communications around a distributed network of relays run by volunteers all around the world: it prevents somebody watching your Internet connection from learning what sites you visit, and it prevents the sites you visit from learning your physical location. Unfortunately, some users have misused such networks — under the cover of anonymity, users have repeatedly defaced popular websites such as Wikipedia. Since web-site administrators cannot blacklist individual malicious users' IP addresses, they blacklist the entire anonymizing network. Such measures eliminate malicious activity through In Anonymizing networks clients are not bounded to share their IP Address to ensure their privacy but some miscellaneous clients misuse this capability of networking under this anonymous cover by impairing well known websites such as Amazon. Anonymizing networks at the cost of denying anonymous access to behaving users. In other words, a few "bad apples" can spoil the fun for all. (This has happened repeatedly with Tor).

There are several solutions to this problem, each providing some degree of accountability. In pseudonymous credential systems users log into websites using pseudonyms, which can be added to a blacklist if a user, misbehaves. Unfortunately, this approach results in pseudonymity for all users, and weakens the anonymity provided by the anonymizing network. Anonymous credential systems employ group signatures. Basic group signatures allow servers to revoke a misbehaving user's anonymity by complaining to a group manager. Servers must query the group manager for every authentication, and thus lacks scalability. Traceable signatures allow the group man-ager to release a trapdoor that allows all signatures generated by a particular user to be traced; such an approach does not provide the backward unlinkability that we desire, where a user's accesses before the complaint remain anonymous. Backward unlinkability allows for what we call subjective blacklisting, where servers can blacklist users for whatever reason since the privacy of the blacklisted user is not at risk. In contrast, approaches without backward unlinkability need to pay careful attention to when and why a user must have all their connections linked, and users must worry about whether their behaviours will be judged fairly. Subjective blacklisting is also better suited to servers such as Wikipedia, where misbehaviours such as questionable edits to a webpage, are hard to define in mathematical terms. In some systems, misbehaviour can indeed be defined precisely. For instance, double-spending of an "e-coin" is considered misbehaviour in anonymous e-cash systems following which the offending user is deanonymized. Unfortunately, such systems work for only narrow definitions of misbehaviour — it is difficult to map more complex notions of misbehaviour onto "double spending" or related approaches. Scheme takes the server about one millisecond per authentication, which is several thousand times faster than VLR. We believe these low overheads will incentivize servers to adopt such a solution when weighed against the potential benefits of anonymous publishing (e.g., whistle-blowing, reporting, anonymous tip lines, activism, and so on.).

II. SUMMARY

A nymble is a pseudo-random number, which serves as an identifier for a particular time period. Nymbles (pre-sented by a user) across periods are unlinkable unless a server has blacklisted that user.The PM issues pseudonyms to users. A pseudonym pnym has two components nym and mac: nym is a pseudo-random mapping of the user's identity (e.g., IP address),7 the linkability window w for which the pseudonym is valid, and the PM's secret key nymKeyP ; mac is a MAC that the NM uses to verify the integrity of the pseudonym.

To limit the number of identities a user can obtain (called the Sybil attack), the Nymble system binds nymbles to resources that are sufficiently difficult to obtain in great numbers. 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.

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.

Nymble tickets are bound to specific time periods, time is divided into linkability windows of duration W, each of which is split into L time periods of duration T (i.e., W = L∗T). If a user misbehaves, the server may link any future connection from this user within the current linkability window (e.g., the same day).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 + 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.

In system, the user can download the server's blacklist and verify her status. If blacklisted, the user disconnects immediately.A thorough protocol redesign has also resulted in several optimizations. Eliminated blacklist version numbers and users do not need to repeatedly obtain the current version number from the NM. Instead servers obtain proofs of freshness every time period and users directly verify the freshness of blacklists upon download based on a hash-chain approach, the NM issues lightweight daisies to servers as proof of a blacklist's freshness, thus making blacklist updates highly efficient. Also, instead of embedding seeds, on which users must perform computation to verify their blacklist status, the NM now embeds a unique identifier nymble, which the user can directly recognize. Figure 1 show basic architecture of Nymble. It includes pseudonym manager, Nymble manager, intended server and user. The figure shows interaction between these entities.

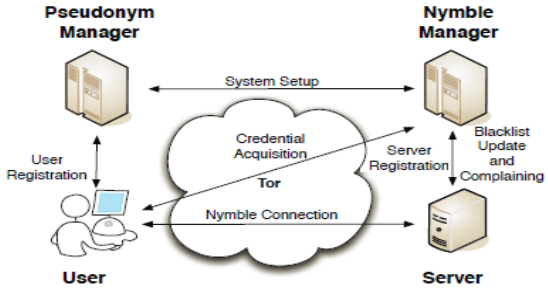

Fig. 1. The Nymble system architecture showing the various modes of interaction

Nymble aims for four security goals. (1)Blacklistabilityassures that any honest server can indeed block misbehaving users. (2) Non-frameability guarantees that any honest user who is legitimate according to an honest server can nymble-connect to that server (3) Rate-limiting assures any honest server that no user can successfully nymble connect to it more than once within any single time period. (4) 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.

Nymble setup stages[10] are:(1) System setup (2) Server registration (3) User registration (4) Credential acquisition (5) Nymble-connection establishment (6) Service provision and access logging (7) Auditing and filing for complaints (8) Blacklist update (9) Periodic update. We will study each stage later in this paper.

Nymble is implemented as a C++ library along with Ruby and JavaScript bindings. One could, however, easily compile bindings for any of the languages (such as Python, PHP, and Perl) supported by the Simplified Wrapper and Interface Generator (SWIG) for example. We utilize Open SSL for all the cryptographic primitives. SHA-256 is used the cryptographic hash functions; HMAC-SHA-256 for the message authentication MA; AES-256 in CBC-mode for the symmetric encryption Enc; and 2048-bit RSASSA-PSA for the digital signatures.

III. REVIEW

In general, each structure (Pseudonyms, Seeds and nymbles, Nymble tickets and credentials, Blacklists, Complaints and linking tokens) grows linearly as the number of entries increases. Credentials and blacklist update requests grow at the same rate because a credential is a collection of tickets which is more or less what is sent. Nymble construction has Blacklistability, Rate-limiting, Non-frameability and Anonymity pro-vided that the trust assumptions hold true, and the cryptographic primitives used are secure.

IP-address blocking by picking IP addresses as the resource for limiting the Sybil attack, current implementations closely mimics IP-address blocking employed by Internet services. There are, however, some inherent limitations to using IP addresses as the scarce resource. If a user can obtain multiple addresses she can circumvent both nymble-based and regular IP-address blocking. Subnet-based blocking alleviates this problem.

Other resources Users of anonymizing networks would be reluctant to use resources that directly reveal their identity (e.g., passports or a national PKI). Email addresses could provide more privacy, but provide weak blacklistability guarantees because users can easily create new email addresses. Other possible resources include client puzzles and e-cash, where users are required to perform a certain amount of computation or pay money to acquire a credential. These approaches would limit the number of credentials obtained by a single individual by raising the cost of acquiring credentials.Server-specific linkability windows an enhancement would be to provide support to vary T and L for different servers. Nymble system does not support varying linkability windows, but does support varying time periods. This is because the PM is not aware of the server the user wishes to connect to, yet it must issue pseudonyms specific to a linkability window. We do note that the use of resources such as client puzzles or e-cash would eliminate the need for a PM, and users could obtain Nymbles directly from the NM. In that case, server-specific linkability windows could be used.

IV. COMPARATIVE STUDY

This system can be compared in different ways.

A. Nymble system

As we know system consists of 9 stages. First let's see what each stage and then we will see what algorithms are used in each stage and what are advantages and disadvantage of each stage. Table 1 shows this comparative study of stages in nymble system. Each stage uses one or more algorithm to serve that stage's purpose, and makes whole system as one unit to give desired output.

1. System setup

During setup, the NM and the PM interact as follows.

1) Both of the systems initialise their state and refresh their memory values.

2) NM executes message authentication – key generation in order to generate a unique key to encrypt and decrypt the pseudonym.

3) Once the keys are exchanged the system setup phase ends.

2. Server registration

In order to participate in the nymble system, the servers must register their server ids and their names along with any of the linked server names if present. The servers register all the details with the NM. The NM acknowledges the receipt of the details along with the encryption key that will be used to encrypt the session ids.

3. User registration

The protocol between user and pseudonym is as follows.

1) The user using its user id such as IP address contacts the PM.

2) The PM verifies its id and issues the pseudoname by using the algorithm pseudorandom generator.

3) Then the user can contact the NM using the encrypted pseudoname and server name for further access to server

4. Credential acquisition

To establish a Nymble-connection to a server, a user must provide a valid ticket, which is acquired as part of a credential from the NM. To acquire a credential for server sid during the current linkability window, a registered user initiates a type-Anon channel to the NM, followed by the Credential Acquisition protocol approval.

5. Nymble Connection Establishment

To establish a connection to the server the user must first contact the NM using its pseudoname and desired server name. The following steps are carried out during connection establishment.

1) Blacklist validation: The server sends the blacklist status to the user along with its signing keys. The signing keys are produced by the message authentication code. The verifying algorithms are used to verify the blacklist status that was updated by the NM.

2) Ticket examination: The NM before issuing the nymble session id verifies the pseudoname if it was signed by the PM. The keys which were shared during the system setup is used to verify the pseudoname.

6. Service provision and access logging

If both the user and the server terminate with success in the Nymble-connection Establishment described above, the server may start serving the user over the same channel. The server records ticket and logs the access during the session for a potential complaint in the future.

7. Auditing and filing for complaints

If both the user and the server terminate with success in the Nymble-connection Establishment described above, the server may start serving the user over the same channel. The server records ticket and logs the access during the session for a potential complaint in the future. If at some later time the server desires to blacklist the user behind a Nymble-connection, during the establishment of which the server collected ticket from the user, the server files a complaint by appending ticket to cmplnt-tickets in its sever state. Filed complaints are batched up. They are processed during the next blacklist update.

8. 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). The procedure for updating blacklists (and their certificates) differs depending on whether complaints are involved. When there is no complaint (i.e., the server's cmplnt-tickets is empty), blacklists stay unchanged; the certificates need only a "light refreshment." When there are complaints, on the other hand, new entries are added to the blacklists and certificates need to be regenerated. So now update has 2 options.

1) Without complaints
2) With complaints

Each option follow has to follow different steps.current implementation employs "lazy" update: the server updates its blacklist upon its first Nymble-connection establishment request in a time period.

9. Periodic update

At the end of a given time period the servers and NM performs a refresh in order to clear all the mentioned blacklist members. During the performance of refresh all the users who were blacklisted is removed. This is done in order to include the concept of what is called as "forgiveness time".

So now we have idea what each stage does and contributes to nymble system. Now we will see a table1 which consists of these stages and algorithms for better understanding.

B. Different models

Now after studying Nymble system, we know how basically it works. This is simplest approach to blacklist misbehaving users in anonymizing networks. It has got 9 stages in its construction .The stages name are listed earlier in paper. In each stage one or more algorithms is implemented in each stage to achieve its purpose(goal).So Nymble is basic system to block users who misbehaves and simplest approach to implement in reality. As we know NM is given the right to blacklist the misbehaved clients for some default period of time. So this is very basic model to restrict misbehaving clients that has aim is to make the servers down or deface it.

Table1 stages in nymble

Sr No.	Stage	Algorithm Used	Advantage	Disadvantage	Conclusion
1	System setup	NMInitState	The NM publishes keys in a way that the users in Nymble can obtain it and verify its integrity at any time.	Have to Generate Verification key every time before proceed further.	Key is generated for users in order to connect to Anonymizing network.
2	Server registration	NMRegisterServer	The NM creates an empty blacklist for the server, signs it and also Instantiates a daisy as a proof of freshness of the blacklist.	We need to check the time period when the blacklist was last updated to the current time period at time of registration.	Empty Blacklist is created to see freshness of user at start up.
3	User registration	UserVerifyBL	Checks the Freshness and integrity of user using downloaded server blacklist.	Every time checks user identity for authentication.	Blacklist is checked to see freshness of user.
4	Credential acquisition	NMVerifyPseudonym , NMCreateCredential	It creates the list of nymble tickets for the user and signs every ticket using a ring signature scheme that allows other NMs and all servers to verify it.	For establishing each new Nymble connection to a server, a user must provide a valid ticket, which is acquired as part of a credential from the NM.	Ticket is generated for user from NM so it can use that ticket to access different servers in network.
5	Nymble connection establishment	ServerVerifyTicket , ServerLinkTicket	The server checks freshness and validity of ticket & also checks whether this ticket is already linked or not.	To establish a connection with a server, each user has to pass through 3 phases-Blacklist validation, Privacy check & Ticket Examination.	Server does verify user integrity which is time consuming process.

6	Service provision and access logging	ServerVerifyTicket	The server records ticket and logs the access during the session for a potential complaint in the future.	Time consuming as it checks logs and records for each complaint.	Server takes ticket of user and serves to user and also creates log this user with its ticket in its database.
7	Auditing and filing for complaints	ServerLink Ticket	The server files a complaint & filed complaints are batched up.	Filed complaints are processed during the next blacklist update which is time gaining process.	Server stores Information about complaints in files in next cycle.
8	Blacklist update	NMHandle Complaints, NMComputeBLUpdate, NMComputeSeeds	The server needs to provide the user with its blacklist certificate for the current time period & to blacklist the misbehaving users by processing the newly filed complaints.	In this multiple updates within a single time period are disallowed (otherwise servers could send users stale blacklists).	Blacklist is getting updated but need to care we don't give simultaneous commands to update.
9	Periodic update	NMSignBL	prepares the linking-token-list for the new time period	At the end of each time period it won't give last updated current likability window with server.	Token list updated for system.

There is Improvised Nymble of basic Nymble model for blocking misbehaving clients in anonymizing networks. It improves basic Nymble model example: In existing basic Nymble system, the Nymble Manager blocks misbehaving clients temporarily for a default period of time, instead of doing that this model uses pseudotracker to track client's rating. If a client misbehaves /impairs a server, server would complain it to NM using the Access Ticket and NM would in turn request the Pseudonym Manager (PM) to find out the client's rating depending on its pseudonym.PM would ask Pseudotracker to check the client's rating and return it to PM.PM will give the rating value from PT to NM. In the end, NM would blacklist/forgive the client depending on its criticality, example: For a newly misbehaved client, rating will be higher say 3.As and when the client will misbehave with different servers the rating will get deteriorated consequently. The right to forgive/blacklist client is given to NM. So this gives idea about the new improvised Nmyble model. In this Nymble Manager (NM) is replaced with Extended Nymble Manager (ENM). A Pseudotracker(PT) is component that is added in basic Nymble model. It interacts with A Nymble Manager (NM).A Pseudotracker is maintained to keep the track of client's rating anonymously by maintaining the client's login details (resources) and the respective rating. Tokenizer is used to give token to user who wants to access server data.Now having idea about this model, let us see how server grants access to user.it consist of ordered steps. (1) User gives ip to tokenizer. (2) tokenizer grants token to user. (3) User interacts with ENM using token. (4) ENM grants ticket-access to if user's rating is greater than threshold value. (5) User accesses the server. This is how this model works.

So after having idea how exactly this two model works. The comparison we can do based on security goals of that is provided by Nymble. These goals are important parameters of system such as Nymble as it provides security to clients. Table 2 shows comparison analysis of these two models. Comparison gives overall idea about both models and its efficiency.

Table 2 Comparison analysis

Parameter	Basic	Improvised
Blacklisting methodology	Temporary	permanent after reaching threshold
Anonymity	It do not ensure anonymity	ensures anonymity
No. of token ticket generation	infinite	Limited tickets 10
Blacklisting of user	User have to download tickets from server	NM notifies the user after permanent blacklisting

This table can be looked as difference between two models also. As it is clear from table that improvised model of Nymble is much better to use and implement in anonymizing networks and it also gives better security but we should not forget it was basic model that has started this new and improvised extensions at it. Thus basic model serves as basic framework to blacklist users

C. Different systems based on Nymble

Nymble[1] was the first anonymous blacklisting scheme. In Nymble.In addition to the SP and the user, thereare two Trusted Parties, the Pseudonym Manager (PM) and the Nymble Manager (NM).Nymble uses an authenticated symmetric encryption scheme E, a pseudorandom function F, a message authentication code MAC and two cryptographic hash functions (modeled as random oracles), f and g; there are two secret keys KP and KN known to the PM and NM, respectively, additionally, the MAC key kpn is shared by the PM and NM and MAC Key kns is shared by the NM and SP. Nymble divides time into "linkability windows," during which
a user's actions can be linked together and these are then divided further into w "time periods". At the beginning of each linkability window d, the user connects directly to the PM to request a pseudonym p= FKP (d,uid); t= MACkpn(p). The user then connects anonymously to the NM, sending p,t; if the tag t is correct, the NM forms a sequence of w + 1 seeds s0 = FKN(p,d), si = f(si-1); tokens ti = g(si); and ciphertexts ci = EKN(t0,si). The NM gives the user nymbles vi = (Ii, ti, ci,MACkns(i, ti, ci)). Then at the i-th time period, the user connects anonymously to the SP, checks that t0 is not blacklisted, and provides vi; the SP grants access if the MAC tag is correct and ti is not blacklisted. To complain, the SP sends vi to the NM, who decrypts ci to get t0; si, and computes ti+1,,,,,,tw and sends these and the "canonical nymble" t0 to the SP to add to the blacklist.

But there are few problems here. There are four possible collusive scenarios between a PM, NM and SP. First, the PM and NM can collude to learn which users connect to which SPs. Second, the NM and SP can collude to link all of a user's actions within a single linkability window. Third, the PM, NM, and SP can all collude together to deanonymize all of the user's activities across linkablity windows. The final scenario involving the PM and SP is not privacy threat to in nymble.

Then came a system based on Nymble named as Nymbler. In Nymbler [18], the PM is replaced by a Credential Manager (CM), who issues an anonymous credential on a secret xuid to each user. The user then uses this credential to create his own series of seeds and tokens, with s0 =h^xuid using f(x) = x^2 mod n, and g(x) = c^x over a trapdoor discrete logarithm group chosen by the NM. The user obtains blind signatures l1,l2,..,lw on the tokens t1,.., tw from the NM, using efficient zero-knowledge proofs to show that they are correctly formed. The SP, on receiving vi = (ti , li) can check the signature, and the NM can extract a seed from ti = ci by computing the discrete logarithm .The use of blind signatures prevents the NM and CM from colluding to link users to SPs; the use of anonymous credentials prevents the NM, CM, and SP from colluding to deanonymize users. The disadvantage is that there is costly computation using the trapdoor.

After Nymbler system came named as Jack[19]. Jack follows Nymbler in replacing the PM with a CM that issue credentials on a secret xuid. The user creates her own nymbles by encrypting a pseudonym hxuid under the NM's public key; the SP maintains a cryptographic accumulator of blacklisted pseudonyms. When the user connects to the SP, she presents her encrypted pseudonym along with a proof of correctness the pseudonym corresponds to the xuid in her credential, is encrypted correctly, and is not in the accumulator. To block a user, the NM decrypts the pseudonym and the SP adds it to the accumulator. As in Nymbler, the use of anonymous credentials prevents deanonymization or linking across linkability windows, and since the user creates nymbles noninteractively, the NM and CM cannot collaborate to link users to SPs.

Then the system came that uses blind signature knows as Bnymble. BNymble[14] uses Chaum's blind signature scheme [20]. In this scheme, the signer has public key N, an RSA modulus, and secret key d =3^-1 mod k(N). It utilize a cryptographic hash function H:M-> Zn modeled as a random oracle. When a user wishes to obtain a blinded signature on the message x €M, she picks r €R Zn and hands b= H(x)r^3 mod N to the signer, who returns k = b^1/3 mod N = H(x)^1/3r mod N. Finally, the user computes o=k/r = H(x)^1/3 mod N. It is easy to see that signing transcripts (b,k) are information-theoretically unlinkable to the signatures (x, H(x)^1/3 mod N).The disadvantage is that proved that it is infeasible to create n + 1 valid signatures from n queries under the one-more RSA inversion problem.

Now after having normal idea about these systems, we will compare efficiency of these systems in terms of basic cryptographic operations required of the users, NM, PM, and SP in each of the systems. Table 3 shows these costs. User registration in BNymble is obviously the most expensive phase, but it is also the least executed protocol - occurring once per linkability window.

Table 3 Cost of cryptographic operations in each "nymble-like" anonymous blacklisting system

	Nymble	BNymble	Jack	Nymbler
User Registration (ms)	0.0008	0.70	9.12	9.12
Nymble acquisition (ms)	0.0027	0.0027	264	649
Nymble veri_cation (ms)	0.0006	0.0006	208	0.0011

All times are measured in high configuration system that has 12GB ram and quad core processor. Figure 2 shows Total cryptographic cost of user registration, nymble acquisition and nymble verification as a function of number of time periods per linkability window. With one week linkability windows and 5-minute time periods, the total cost of BNymble is only 11% higher than Nymble. Basically Figure 2 shows how this one-time cost compares to the total cost of authentication for various linkability window sizes.

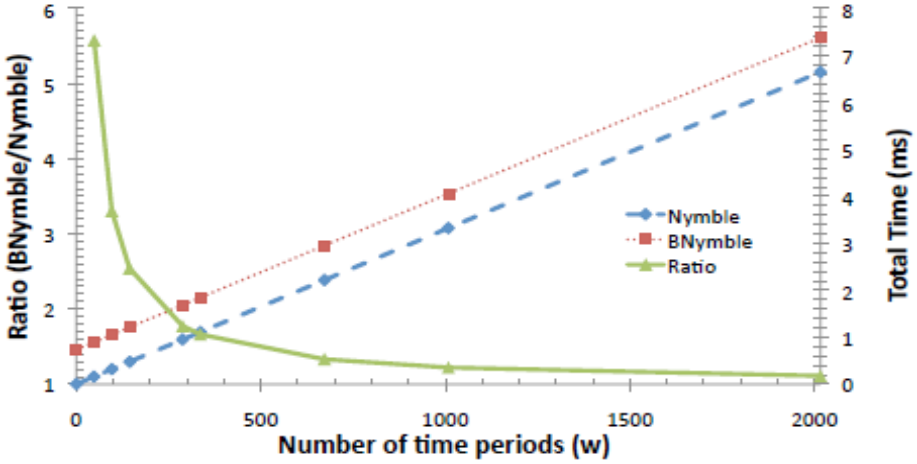

Fig. 2 time comparison of different systems.

V. ADVANTAGES

*1) Blacklisting anonymous users:*It provides a meansby which servers can blacklist users of an anonymizing network while maintaining their privacy.

*2) Practical performance:*This system makes use ofinexpensive symmetric cryptographic operations to significantly outperform the alternatives.

*3) Open-source implementation.*With the goal of contributing a workable system, system is built as an open-source implementation of Nymble, which is publicly available.

VI. DISADVANTAGES

1) This blocking system does not provide security; any users can change IP and enter into network.

2) There is no way to stop anonymity in this system and
Website admins can block only fake IP.

3) Side-channel attacks, while current implementation does not fully protect against side-channel attacks, we mitigate the risks. But there are some various algorithms available in a way that their execution time leaks little information that cannot already be inferred from the algorithms out

4) This Nymble system blacklists the user for some amount of time; we need extra mechanism to blacklist a user permanently in case if needed.

VII. FUTURE SCOPE

In future Nymble system we can have system that finds client mac or physical address and each time the client entry is maintained. When users misbehave in the website, admins can block their mac or physical address. so blocked user can't access internet services. This system will perfectly in LAN have to host this software and IIS server used to store information all the physical address and mac address. The future system will be configured for wireless, WAN, MAN networks and secure the website from anonymity. In future we can expect features like coin recovery [14], identity logging, extended blacklisting, resisting traffic analysis in nymble systems

REFERENCES

[1] P. C. Johnson, A. Kapadia, P. P. Tsang, and S. W. Smith. Nymble:Anonymous IP-Address Blocking. In Privacy Enhancing Technologies,LNCS 4776, pages 113–133. Springer, 2007

[2] C. Cornelius, A. Kapadia, P. P. Tsang, and S. W. Smith. Nymble: Blocking Misbehaving Users in Anonymizing Networks. Technical Report TR2008-637, Dartmouth College, Computer Science,Hanover, NH, December 2008.

[3] M. Bellare and P. Rogaway. Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols. In Proceedings of the 1st ACM conference on Computer and communications security, pages 62–73. ACM Press, 1993

[4] [4]M. Bellare, R. Canetti, and H. Krawczyk. Keying Hash Functions for Message Authentication. In CRYPTO, LNCS 1109, pages 1–15. Springer, 1996

[5] [5]M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A Concrete Security Treatment of Symmetric Encryption. In FOCS, pages394–403, 1997

[6] [6]S. Goldwasser, S. Micali, and R. L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM J. Comput., 17(2):281–308, 1988

[7] [7]S. Even, O. Goldreich, and S. Micali. On-Line/Off-Line Digital Schemes. In CRYPTO, LNCS 435, pages 263–275. Springer, 1989

[8] [8]A. Juels and J. G. Brainard. Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks. In NDSS. The Internet Society, 1999

[9] [9] Anonymizing Networks, by Peng Deng,University of Melbourne

[10] [10] Nymble: Blocking Misbehaving Users in Anonymizing Networks by Patrick P. Tsang, Apu Kapadia, Member, IEEE, Cory Cornelius, and Sean W. Smith

[11] [11]http://en.wikipedia.org/wiki/Tor_%28anonymity_network%29

[12] [12]http://www.theguardian.com/technology/2013/nov/05/tor-beginners-guide-nsa-browser

[13] [13]http://www.onion-router.net/

[14] [14] BNymble More anonymous blacklisting at almost no cost, By Peter Lofgren and Nicholas,Hopper,University of Minnesota

[15] [16] http://en.wikipedia.org/wiki/Blacklisting

[16] Towards Efficient Traffic-analysis Resistant Anonymity Networks by Stevens Le Blond David Choffnes Wenxuan Zhou

[17] [18] Making a Nymbler Nymble using VERBS by Ryan Henry, Kevin Henry, and Ian Goldberg, University of Waterloo

[18] [19] Jack: Scalable Accumulator-based Nymble System by Zi Lin and Nicholas Hopper, University of Minnesota

[19] [20] http://en.wikipedia.org/wiki/Blind_signature

[20] [21] Tsang, P.P., Au, M.H., Kapadia, A., Smith, S.W.: BLAC: Revoking Repeatedly Misbehaving Anonymous Users Without Relying on TTPs. Tech. rep., Dartmouth Computer Science TR2008-635 (2008

