The International Journal Of Engineering And Science (IJES)
||Volume||2 ||Issue|| 2 ||Pages|| 208-213 ||2013||
ISSN: 2319 – 1813 ISBN: 2319 – 1805

Delimiting Malicious Users In Pseudo Networks

1,Mahesh Uddandam, 2,T.V.Sai Krishna
1,M.Tech (CSE) Student, Dept of CSE, QISCET
2,Assoc Professor, Dept of CSE, QISCET

Abstract

The main aim of this Project is to reduce Malicious Users, which can ingress the blocked WebPages via Pseudo Networks. Now a day's lot of people misuses the attractive sites like Face book, Twitter, Orkut. etc by using client's IP address. So the site owner can block the client's IP address from the server, but it segmenting IP address is not an easier way, because the fake routes through an pseudo networks, as a sequel admin congest all familiar nodes of pseudo networks, refuse access to all there by unknown users whether misbehaving or not. So to acquire a solution, we introduce a new technique called Nymble, where servers delimit malicious users able to restricting users, without affecting their ambiguity & this system accurately finding malicious users in nameless network and also maintained restricting user's details in server.

***Key Words**: Malicious Users, Pseudo Network, Unlinkability.*

Date Of Submission: 19, February, 2013 ⟺ Date Of Publication: 28, February 2013

I. Introduction

In order to hide a client's IP address anonymizing networks like Tor [12] route traffic through independent nodes in separate administrative domains. Some users, however, have misused such networks under the cover of anonymity; they have repeatedly defaced popular Web sites such as Wikipedia. As administrators cannot block individual users' IP addresses, they resort to Blacklisting the entire anonymizing network. However, such methods though eliminate malicious activity through anonymizing networks but they also deny anonymous access to behaving users. (A case repeatedly observed with Tor.1). In a pseudonymous credential system [13], users log into Web sites using pseudonyms, which can be blocked if in case a user misbehaves. However, this method may results in pseudonym for all users, thereby dampening the Anonymity provided by the anonymizing network. Anonymous credential systems [10], [12], employ group signatures. On the other hand, basic group signatures [2], [7], [14], allow servers to annul a misbehaving user's anonymity by complaining about it to a group manager. Servers must contact the group Manager for every authentication, and thus, this method lacks scalability. Traceable signatures, Help the group manager, which then release a trapdoor allowing all signatures generated by a particular user to be traced. Even though, using such an approach does not provide the necessary backward unlinkability that we desire, a user's accesses before the complaint always remain anonymous. These methods hold true for only a few definitions of misbehavior it is quite arduous to map more complex definitions of misbehavior with related approaches [4] With dynamic accumulators [11] a canceling operation might result in a new accumulator and public parameters for the group, and making it mandatory to update all other existing users' credentials, thus making it impractical. Verifier-local revocation (VLR) [3], [8], [9] overcomes this by requiring the server ("verifier") to perform only local updates during revocation.

1.1 Our solution

We close to a safe system called Nymble, which provide all the following property: unsigned authentication, backward unlink ability, biased blacklisting, fast authentication speed rate-limited anonymous associations, revocation audit ability (where users can verify whether they have been blacklisted), and also addresses the Sybil attack to make its consumption realistic.Our scheme ensures that users are aware of their blacklist status before they present a Nymble, and separate straight away if they are blacklisted. Even though 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 nameless users regardless of their Servers can consequently blacklist nameless users not including knowledge of their IP addresses while allowing behaving users to connect namelessly.

Our research makes the following contributions:

- Blacklisting unsigned users**.** We provide a means by which servers can blacklist clients of an anonymizing system while maintaining their privacy.
- Realistic performance**.** Our procedure makes use of identity certificates, and trusted hardware. We address economical symmetric cryptographic operations to the practical issues related with resource-based blocking extensively outperform the alternatives.
- Open-source achievement**.** With the goal of causative a practical system, we have built an open-source achievement of Nymble, which is publicly available.5 we provide concert statistics to show that our system is indeed sensible

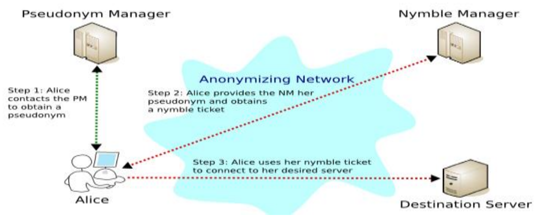
Pseudonym Manager
Nymble Manager
Anonymizing Network
Step 1: Alice contacts the PM to obtain a pseudonym
Step 2: Alice provides the NM her pseudonym and obtains a nymble ticket
Step 3: Alice uses her nymble ticket to connect to her desired server
Alice
Destination Server

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

Some of the authors of this paper have available two unsigned validation schemes, BLAC [5] and PEREA [6], which do away with the need for a trusted third party for revoking clients. Nymble thus represents a practical solution for blocking mischievous clients of anonymizing communications. We note that a comprehensive version of this article is obtainable as a technical report [15].

II. Overview To Nymble

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.

2.1 Resource-Based Blocking

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. 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 address the practical issues related with resource-based blocking in Section 8, and suggest other alternatives for resources. 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

2.2 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

2.3 The Nymble Manager

To provide the requisite cryptographic protection And security properties, the NM encapsulate 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.

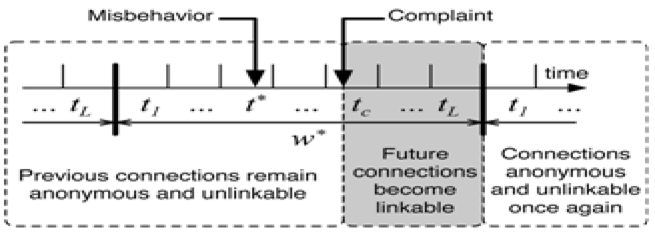
Misbehavior
Complaint
time
… t_L t_1 … t^* … t_c … t_L t_1 …
w^*
Previous connections remain anonymous and unlinkable
Future connections become linkable
Connections anonymous and unlinkable once again

Fig. 2. The life cycle of a misbehaving user. If the server complains in Time period t_c about a user's connection in t^*, the user becomes linkable Starting in t_c. The complaint in t_c can include nymble tickets from only t_{cc-1} and earlier.

2.4 Time

Nymble tickets are bound to specific time periods. As illustrated in Fig. 2, time is divided into linkability windows of duration W, each of which is split into L time periods of duration T (i.e., $W = L^*T$). We will refer to time periods and linkability windows chronologically as $t_1, t_2, \ldots, t_L$ and $w_1, w_2, \ldots$, 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 misbehaviors from a particular user before he or she is blocked.

2.5 Blacklisting a User

Users who make use of anonymizing networks Expect their connections to be anonymous. If a Server obtains a seed for that user, however, it can Link that user's subsequent connections. It is of Utmost importance, then, that users be notified of Their blacklist status before they present a nymble Ticket to a server. In our system, the user can Download the server's blacklist and verify her Status. If blacklisted, the user disconnects Immediately. 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, and while it is Possible to modify our system to support subnet based Blocking, new privacy challenges emerge; A more thorough description is left for future Work

2.6 Notifying the User of Blacklist Status

Users who make use of anonymizing networks Expect their connections to be anonymous. If a server obtains a seed for that user, however, it can link that user's subsequent connections. It is of utmost importance then that users be notified of their blacklist status before they present a nimble ticket to a server. In our system, the user can download the server's blacklist and verify her status. If blacklisted, the user disconnects immediately.

III. Our Nymble Construction

3.1 System Setup

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

1. The NM executes NMInitState() (see Algorithm 1) and initializes its state nmState to the algorithm's output.
2. The NM extracts $macKey_{NP}$ from nmState and sends it to the PM over a type-Auth channel. $macKey_{NP}$ 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 $nymKey_P$ by running Mac.Key-Gen() and initializes its state pmState to the pair ($nymKey_P$, $macKey_{NP}$) .
4. The NM publishes $verKey_N$ in nmState in a way that the users in Nymble can obtain it and verify its integrity at any time (e.g., during registration).

Algorithm 1. NMInitState

Output: nmState $\in S_N$

```
1: $macKey_{NP}$ := Mac:KeyGen()
2: $macKey_N$ := Mac:KeyGen()
3: seedKeyN := Mac:KeyGen()
4: ($encKey_N$; $decKey_N$ := Enc:KeyGen()
5: ($signKey_N$; $verKey_N$) := Sig:KeyGen()
6: keys := ($macKey_{NP}$ , $macKey_N$, $seedKey_N$,
7: $encKey_N$, $decKey_N$, $signKey_N$, $verKey_N$)
8: nmEntries := $\phi$
9: return nmState := (keys,nmEntries)
```

3.2 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 linkability window.

1. The NM makes sure that the server has not already registered: If (sid, .,.) $\in$ nmEntries in its nmState, it terminates with failure; it proceeds otherwise.
2. TheNM reads the current time period and linkability window as t_{now} and w_{now}, respectively, and then obtains a svrState by running (see Algorithm 2)

   $NMRegisterServer_{nmState}(sid,t_{now},w_{now})$.
3. The NM appends svrState to its nmState, sends it to the Server, and terminates with success.
4. The server, on receiving svrState, records it as its state, and terminates with success.

```
Algorithm 2. NMRegisterServer
Input: (Sid, t, w) $\in H*N^2$
Persistent state: nmState $\in S_N$
Output: svrState $\in S_S$
1: (keys,nmEntries) :=nmState
2: $macKey_{NS}$ := Mac:KeyGen()
3: $daisy_L \in R^H$
4: $nmEntries^{'}$ := nmEntries|| (sid, $macKey_{NS}$, $daisy_L$, t)
5: nmState := (keys,$nmEntries^{'}$)
6: target := $h^{(L-t+1)}$(daisyL)
7: blist := $\phi$
8: cert := $NMSignBL_{nmState}$(sid, t, w, target, blist)
9: svrState := (sid, macKeyNS, blist, cert, $\phi$, $\phi$, $\phi$, t)
10: return svrState
```

In svrState, macKeyNS is a key shared between the NM and the server for verifying the authenticity of nimble tickets; $time_{lastUpd}$ indicates the time period when the blacklist was last updated, which is initialized to t_{now}, the current time period at registration.

3.3 User Registration

A user with identity uid must register with the PM once in each linkability window. To do so, the user initiates a type- Basic channel to the PM, followed by the User Registration protocol described below.

1.The PM checks if the user is allowed to register. In our current implementation, the PM infers the registering user's IP address from the communication channel, and makes sure that the IP address does not belong to a known Tor exit node. If this is not the case, the PM terminates with failure.

2. Otherwise, the PM reads the current linkability window as w_{now}, and runs

pnym := $PMCreatePseudonym_{pmState}$(uid,wnow).

The PM then gives pnym to the user, and terminates with success.

3. The user, on receiving pnym, sets her state usrState to (pnym, ϕ), and terminates with success.

3.4 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 below.

1. The user extracts pnym from usrState and sends the Pair (pnym, Sid) to the NM.

2. The NM reads the current linkability window as w_{now}. It makes sure the user's pnym is valid: If

$NMVerifyPseudonym_{nmState}(pnym,w_{now})$

returns false, the NM terminates with failure; it proceeds otherwise.

3.The NM runs $NMCreateCredential_{nmState}(pnym, sid, w_{now})$, This returns a credential cred. The NM sends cred to the user and terminates with success.

4. The user, on receiving cred, creates usrEntry:=(sid, cred, false), appends it to its state usrState, and terminates with success.

3.5 Nymble Connection Establishment

To establish a connection to a server sid, the user initiates a type-Anon channel to the server, followed by the Nymble connection establishment protocol described below.

3.6 Privacy Check

Since multiple connection establishment attempts by a user to the same server within the same time period can be linkable, the user keeps track of whether she has already disclosed a ticket to the server in the current time period y maintaining a Boolean variable ticketDisclosed for the server in her state. Furthermore, since a user who has been blacklisted by a server can have her connection establishment attempts linked to her past establishment, the ser must make sure that she has not been blacklisted thus far. Consequently, if ticketDisclosed in usrEntries[sid] in the user's usrState is true, or UserCheckIfBlacklisted$_{\text{usrState}}$(sid, blist) = true, then it is unsafe for the user to proceed and the user sets safe to false and terminates the protocol with failure.

3.7. Ticket Examination

1. The user sets ticketDisclosed in usrEntries[sid] in usrState to true. She then sends hticketi to the server, where ticket is ticket[$t_{now}^{(U)}$] in cred in usrEntries[sid] in usrState. Note that the user discloses ticket for time period $t_{now}^{(U)}$ after verifying blist's freshness for $t_{now}^{(U)}$. This procedure avoids the situation in which the user verifies the current blacklist just before a time period ends, and then presents a newer ticket for the next time period.
2. On receiving hticketi, the server reads the current time period and linkability window as $t_{now}^{(S)}$ and $w_{now}^{(S)}$, respectively. The server then checks that:

- ticket is fresh, i.e., ticket $\notin$ slist in server's state.
- ticket is valid, i.e., on input ($t_{now}^{(S)}$, $w_{now}^{(S)}$, ticket), the algorithm ServerVerifyTicket returns true. (See Algorithm 3.)
- ticket is not linked (in other words, the user has not been blacklisted), i.e.,

ServerLinkTicket$_{\text{svrState}}$(ticket) = false:

(See Algorithm 3.)

3. If any of the checks above fails, the server sends (goodbye) to the user and terminates with failure.Otherwise, it adds ticket to slist in its state, sends (okay) to the user, and terminates with success.
4. On receiving hokayi, the user terminates with success.

```
Algorithm 3. ServerLinkTicket
Input: ticket ∈ T
Persistent state: svrState ∈ S_S
Output: b ∈ {true; false}
1: Extract lnkng-tokens from svrState
2: (., nymble, …) := ticket
3: for all i = 1 to [lnkng-tokens] do
4: if (,., nymble) = lnkng-tokens[i] then
5: return true
6: return false
```

IV. Security Analysis

Our Nymble construction has Blacklistability, Rate-limiting, Nonframeability, and Anonymity provided that the trust assumptions in Section 3.2 hold true, and the cryptographic primitives used are secure.

4.1 Blacklistability

An honest PM and NM will issue a coalition of c unique users at most c valid credentials for a given server. Because of the security of HMAC, only the NM can issue valid tickets, and for any time period, the coalition has at most c valid ickets, and can thus make at most c connections to the server in any time period regardless of the server's blacklisting. It suffices to show that if each of the c users has been blacklisted in some previous time period of the current linkability window, the coalition cannot authenticate in the current time period k^*.Assume the contrary that connection establishment k^* using one of the coalition members' ticket* was successful even though the user was blacklisted in a previous time period k^j. Since connection establishments k^j and k^* were successful, the corresponding tickets ticket$'$ and ticket* must be valid. Assuming the security of digital signatures and HMAC, an honest server can always contact an honest NM with a valid ticket and the NM will successfully terminate during the blacklist update. Since the server blacklisted the valid ticket$'$ and updates it's linking list honestly, the ServerLinkTicket will return fail on input ticket*, and thus, the connection k^* must fail, which is a contradiction.

4.2 Nonframeability

Assume the contrary that the adversary successfully framed honest user i^* with respect to an honest server in time period t^*, and thus, user i^* was unable to connect in time period t^* using ticket* even though none of his tickets were previously blacklisted. Because of the security of HMAC, and since the PM and NM are honest, the adversary cannot forge tickets for user i^*, and the server cannot already have seen ticket*; it must be that ticket* was linked to an entry in the linking list. Thus, there exists an entry (seed*; nymble$^{*)}$ in the server's linking list, such that the nymble in ticket* equals nymble* the server must have obtained this entry in a successful blacklist update for some valid ticketb, implying the NM had created this ticket for some user $i^{\sim}$. If $i^{\sim} \neq i^*$, then user $i^{\sim}$'s seed0 is different from user i^*'s $seed_0$ so long as the PM is honest, and yet the two $seed_0$'s evolve to the same seed*, which contradicts the collision resistance property of the evolution function. Thus, we have $i^{\sim} = i^*$. But, as already argued, the adversary cannot forge i^*'s $ticket_b$, and it must be the case that i^*'s $ticket_b$ was blacklisted before t^*, which contradicts our assumption that i^* was legitimate user in time t^*.

V. Conclusion

We have anticipated and built an ample system, which can be used to add a layer of liability to any openly known anonymizing network. Servers can blacklist mischievous users though maintaining their seclusion, and we show how these properties can be attained in a way that is practical, efficient, and perceptive to requirements of both users and services. We hope that our effort will amplify the conventional acceptance of anonymizing networks such as Tor, which has thus far been entirely restricted by several services because of users who cruelty their anonymity.

References

[1] Patrick P. Tsang, Apu Kapadia, Member, IEEE, Cory Cornelius, and Sean W. Smith "Nymble: Blocking Misbehaviour User In anonymizing Network" Vol No 8, No. 2, Pages 256-269, 2011.

[2] G. Attendees, J. Camelish, M. Joey, and G. Studio. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In CRYPTO, LNCS 1880, pages 255–270. Springer, 2000.

[3] G. Attendees, D. X. Song, and G. Studio. Quasi-Efficient Revocation in Group Signatures. In Financial Cryptography, LNCS 2357, pages 183–197. Springer, 2002.

[4] I. Tarnish, J. Furukawa, and K. Saco, "k-Times Anonymous Authentication (Extended Abstract)," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), Springer, pp. 308-322, 2004

[5] P.P. Tsang, M.H. Au, A. Acadia, and S.W. Smith, "Blacklist able Anonymous Credentials: Blocking Misbehaving Users without TTS," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 72-81, 2007.

[6] P.P. Tsang, M.H. Au, A. Acadia, and S.W. Smith, "PEREA: Towards Practical TTP-Free Revocation in Anonymous Authentication,"Proc. ACM Conf. Computer and Comm. Security, pp. 333-344, 2008.

[7] M. Bellare, H. Shi, and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In CT-RSA, LNCS 3376, pages 136–153. Springer, 2005.

[8] D. Boneh and H. Shacham. Group Signatures with Verifier-Local Revocation. In ACM Conference on Computer and Communications Security, pages 168–177. ACM, 2004.

[9] E. Bresson and J. Stern. Efficient Revocation in Group Signatures. In Public Key Cryptography, LNCS 1992, pages 190–206. Springer, 2001.

[10] J. Camelish and A. Lysyanskaya. An Efficient System for Nontransferable Anonymous Credentials with Optional Anonymity Revocation. In EUROCRYPT, LNCS 2045, pages 93–118. Springer, 2001.

[11] J. Camelish and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In CRYPTO, LNCS 2442, pages 61–76. Springer, 2002.

[12] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second- Generation Onion Router," Proc. Usenix Security Symp., pp. 303- 320, Aug. 2004.

[13] D. Chaum. Showing Credentials without Identification Transfeering Signatures between Unconditionally Unlinkable Pseudonyms. In AUSCRYPT, LNCS 453, pages 246–264. Springer, 1990.

[14] D. Chaum and E. van Heyst. Group Signatures. In EUROCRYPT, pages 257–265, 1991.

[15] C. Cornelius, A. Acadia, 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.

AUTHORS PROFILE

Mahesh Uddandam pursuing M.Tech (CSE) from QIS College of Engineering and Technology, Ongole Affiliated to JNTU, Kakinada. His Interested areas include Secure Computing, Computer Networks and Network Security.

Mr T.V.SaiKrishna currently working as Associate Professor in Dept of CSE. He received his B.Tech from JNTU Hyderabad and M.Tech from JNTU, Anantapur. He had 9 years of Teaching Experience and Life Member of ISTE. He has published several papers at various national and International Journals and Conferences. His Research areas include Image Processing, Data mining and Network Security.