Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 6, June 2013, pg.193 – 199

RESEARCH ARTICLE

Security Refinement in Nymble System

Maheshkumar S. Kamble[1], **Hatkar S. S.**[2]
[1]M. Tech. Student, Dept. of CSE, SGGS IE&T, Nanded, India
[2]Associate Professor, Dept. of CSE, SGGS IE&T, Nanded, India

[1] *maheshkumar.kamble@gmail.com;* [2] *Shubhanand.hatkar@yahoo.com*

Abstract— Anonymity in computer networks empowers users to access internet services anonymously and prevents any tracking or tracing of their identity on the World Wide Web. Many open source applications like Tor provide such anonymity. Traffic analysis and network surveillance are prevented by such type of networks. This facilitates hidden services to users and cover up internet protocol address of anonymous users. Many users take advantage of anonymity and use it for abusive purpose and remain hidden after misbehaving. As a result of this, website administrators block all well-known exit nodes of anonymous network and prevent misbehaved users as well as behaved users from accessing the website. To address this issue we present a Nymble system in which servers can blacklist misbehaving users by preserving user's anonymity.

Key Terms: - anonymity preservation; blacklist; pseudonymity; unlinkability

I. Introduction

Anonymous networks such as Tor [1] over the internet offering a simple layer that identity-sensitive application can use to securely communicate. To hide client's IP address data is wrapped with several layers of encryption, and the network is both distributed and dynamic administrative domains. Tor provides the foundation for a range of applications that allow organizations and individuals to share information over public networks without compromising their privacy. Using Tor protects you against a common form of Internet surveillance known as "traffic analysis". Traffic analysis can be used to infer who is talking to whom over a public network. Knowing the source and destination of your Internet traffic allows others to track your behaviour and interests. Unluckily some of users make use of such networks for abusive purpose such as defacing popular websites such as Wikipedia. To avoid such attack website administrator block the entire anonymous network. Such precaution avoids malicious activity through anonymous networks, but well behaved users suffer because they cannot access website through anonymous networks. Here we can say that, because of some misbehaved users all well behaved users have to suffer.

Several solutions are proposed and implemented to solve this problem such as pseudonym credential systems [6] where pseudonyms are basis for blacklist a misbehaved user over anonymous networks. Unfortunately this approach weakens the anonymity over Tor network because this results in pseudonymity for all users. Another approach is anonymous credential system [5] with group signatures [7] where servers complain to group manager and then it can revoke a misbehaving user's anonymity. This approach lacks in scalability.

II. An Overview to Nymble

Here, we present a Nymble system which uses a novel construction to build mutually unlinkable and verifiable authentication tokens for users of anonymous networks, while empowering service providers with access revocation capabilities comparable to what they have with non-anonymous users. In particular, this scheme implements a privacy-preserving IP address blocking for users who communicates through anonymous networks. User acquires collection of nymbles, a special type of pseudonym to connect to websites. Websites can blacklist user by obtaining a seed for a particular nymble, allowing them to link future nymbles from the same user by making the nymbles which were used before complaints remain unlinkable. Hence, server can blacklist the anonymous user without knowledge of their IP addresses while allowing well behaving users to connect anonymously. In this system, user should be aware of her blacklist status before she communicates with Nymble system and disconnect immediately if blacklisted.

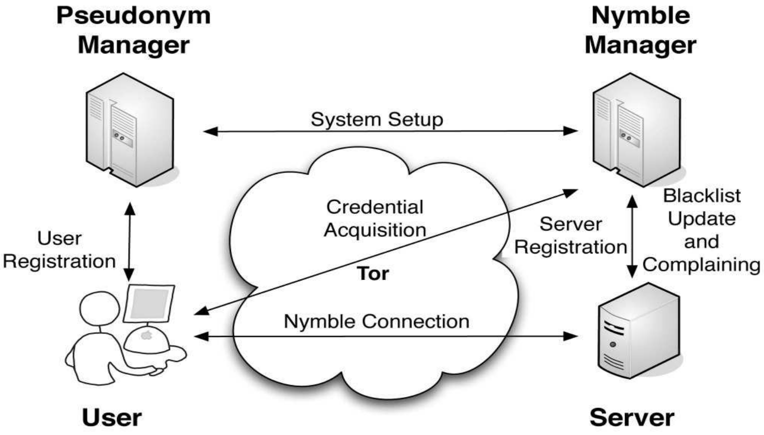


Figure 1. The Nymble system architecture [2]

A. Pseudonym Manager(PM)

User need to contact the pseudonym manager and demonstrate control over a particular resource in order to get its IP-address blocked. The user is required to connect to the PM directly i.e. not through a known anonymous network. Pseudonym Manager has the knowledge about Tor routers and hence it won't accept it if a user tries to connect with it with anonymous network.

The basic idea behind connecting directly with Pseudonym Manager is that, it can identify the IP-address of the user. Pseudonyms are chosen based upon the controlled resource ensuring that the same pseudonym is always issued for the same resource. Pseudonym Manager only knows the IP address-pseudonym pair and hence it does not know the server to which the user wants to connect. User contacts the Pseudonym manager only once per linkability window (e.g. Once a day).

B. Nymble Manager(NM)

After getting the pseudonym from the pseudonym manager, the user connects to the Nymble Manager through anonymous network and requests nymbles for access to a particular server.

Nymbles are generated using the user's pseudonym and the server's identity. Nymble Manager doesn't know anything about the user's identity. It knows only the pseudonym-server pair. Nymble Manager encapsulates nymbles within "Nymble tickets" in order to provide cryptographic protection and security properties.

Nymble tickets are bound to specific time periods. In Nymble system, time is divided into linkability windows of duration W and each w is split into L time periods of duration T, i.e. W=L*T.

III. TERMINOLOGY

In this section, we will discuss some aspects of anonymous communications over computer networks.

A. Address Anonymity

While surfing the web, the computer connects to the target server by contacting the web page with the help of user's IP address and sharing other information like the browser and operating system version. This information can be used to track down the user. However, a certain degree of anonymity can be achieved by using a proxy server and anonymous networks like Tor [1] and Crowds [10]. The proxy server works by redirecting the communication through itself. The browser's IP address is then only shared with the proxy server while the target website only sees the proxy server's information. Tor makes use of Onion Routing for hiding IP address of communicating users while forwarding packets through random set of dedicated routers.

In this way both sender and receiver remains anonymous which is basic need for privacy preservation in shared public network.

B. Unlinkability

This is most important aspect of anonymous communications between end users which assures that two or more related events in an information processing system cannot be related to each other. In other words, a user may make multiple uses of resources or services without others being able to link these uses together. Unlinkability is considered to be a sub property of privacy.

C. Unobservability

Undetectability of an item of interest from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not. Unobservability of an item of interest means undetectability of the item of interest against all subjects uninvolved in it and anonymity of the subject involved in the item of interest even against the other subject involved in that item of interest.

Following TABLE I define the relationship between above mentioned aspects of anonymous communications.

TABLE I

Relationships between aspects		
unobservability	→	anonymity
sender unobservability	→	sender anonymity
recipient unobservability	→	recipient anonymity
relationship unobservability	→	relationship anonymity
sender anonymity	→	relationship anonymity
recipient anonymity	→	relationship anonymity
sender unobservability	→	relationship unobservability
recipient unobservability	→	relationship unobservability
unobservability	→	undetectability

D. Pseudonymity

Pseudonymity means using a pseudonym instead of one's "real" name. Pseudonyms are typically assumed to be the same person or collective working as one entity over time. Most websites have login controls so that a registered username must be unique and that whoever is posting under that username must know the password or have access to the email address controlling the account.

We found that users of a large anonymity network were being denied access to popular internet services as a result of the abuse made possible by strong anonymity. *Pseudonymity* allows clients to obtain and use pseudonym credentials with a minimum of effort, without even installing additional software. It allows service providers to accept these credentials with a minimum of effort which are used to invoke misbehaving user's identity over an anonymous network.

IV. SECURITY MODEL

Nymble system employees following security goals which are responsible for resistance towards coalition attacks and security to anonymity of user.

A. Blacklistability

It assures that any honest server can block misbehaving user in current linkability window.

B. Rate-limiting

It assures that any honest server that no user can successfully connect to it more than once within single time period.

C. Nonframeability

No one can frame a well behaving user for the activity of malicious user.

D. Anonymity

It protects anonymity of honest users regardless of their legitimacy. Server cannot acquire any information over a nymble connection.

V. CRYPTOGRAPHIC PRIMEVAL

Nymble system uses following building blocks to achieve highest degree of network security.

***Cryptographic Hash Functions*: **These functions are typically used to compute a message digest when making a digital signature [12]. Instead of encrypting the whole message with the secret key, only the message digest is encrypted. This is much faster than encryption the complete message.

A hash function compresses the bits of a message to a fixed-size hash value in a way that distributes the possible messages evenly among the possible hash values. A cryptographic hash function does this in a way that makes it extremely difficult to come up with a message that would hash to a particular hash value.

Cryptographic hash functions typically produce hash values of 128 or more bits. This number is vastly larger than the number of different messages likely to ever be exchanged in the world.

***Message Authentication*:** A message authentication code (MAC) [12] is an authentication tag (also called a checksum) derived by applying an authentication scheme, together with a secret key, to a message. Unlike digital signatures, MACs are computed and verified with the same key, so that they can only be verified by the intended recipient.

***Symmetric Key Encryption*:** In this scheme each party has a secret key (code) that it can use to encrypt a packet of information before it is sent over the network to another party. Symmetric-key requires that you know which party will be talking to each other so you can install the key on each one. Symmetric-key encryption is essentially the same as a secret code that each of the two parties must know in order to decode the information. The code provides the key to decoding the message [13].

***Digital Signatures*:** A digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document, and possibly to ensure that the original content of the message or document that has been sent is unchanged [13]. Digital signatures are easily transportable, cannot be imitated by someone else, and can be automatically time-stamped. The ability to ensure that the original signed message arrived means that the sender cannot easily repudiate it later.

A digital signature can be used with any kind of message, whether it is encrypted or not, simply so that the receiver can be sure of the sender's identity and that the message arrived intact. A digital certificate contains the digital signature of the certificate-issuing authority so that anyone can verify that the certificate is real.

A digital code that can be attached to an electronically transmitted message that uniquely identifies the sender. Like a written signature, the purpose of a digital signature is to guarantee that the individual sending the message really is who he or she claims to be. Digital signatures are especially important for electronic commerce and are a key component of most authentication schemes. To be effective, digital signatures must be unforgeable.

VI. IMPLEMENTATION AND EXPERIMENTAL SETUP

We implemented Nymble using Servlets and JSP bindings. Java Servlet technology and Java Server Pages (JSP pages) are server-side technologies that have dominated the server-side Java technology market; they've become the standard way to develop commercial web applications. Java developers love these technologies for myriad reasons, including: the technologies are fairly easy to learn, and they bring the *Write Once, Run Anywhere* paradigm to web applications. More importantly, if used effectively by following best practices, servlets and JSP pages help separate presentation from content. *Best practices* are proven approaches for developing quality, reusable, and easily maintainable servlet- and JSP-based web applications. The advantages of using servlets are their fast performance and ease of use combined with more power over traditional CGI (Common Gateway Interface).

In this section we use following Algorithm1 to run a Nymble System:

```
Algorithm1
    Step 1:  NM calculates $macKey_{NP}$ which is a shared key between NM and PM.
    Step 2:  NM send $macKey_{NP}$ to PM over an authenticated channel.
    Step 3:  Server initiates a connection with NM and send its identity sid to NM.
    Step 4:  NM calculate svrState (State of Server) which contains shared key $macKey_{NS}$ between Server and
        NM and send it to Server.
    Step 5:  Server update its svrState and terminate with success.
    Step 6:  User initiates a connection to PM over an authenticated channel and describe control over IP.
    Step 7:  PM calculates pnym (pseudonym), send it to user and terminate with success.
    Step 8:  User update its usrState (State of User).
    Step 9:  User initiates connection through anonymous network to NM and send (pnym, sid) to NM.
    Step 10: NM calculate cred (credentials), send it to user and terminate with success.
    Step 11: User initiates connection with server through anonymous network and submit credentials to server.
    Step 12: If User misbehaves with Server, Server can blacklist the User with the help of cred and block it for
        given linkability window.
```

We utilize Bouncy Castle Crypto APIs for many cryptographic primitives. Very first version of Nymble system [2] presented by Patric P. Tsang, cryptographic hash functions are implemented using SHA-256. Here, in our implementation we use SHA-3 instead of SHA-256 to implement cryptographic hash functions. As of May

2013, NIST (National Institute of Standards and Technology) has not yet updated the Secure Hash Standard (SHS) for SHA-3. Originally SHA-3 known as Keccak [15] which is a cryptographic hash function designed by Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche,

On October 2, 2012, Keccak was selected as the winner of the NIST hash function competition. Keccak is a family of sponge functions [14]. The *sponge function* is a generalization of the concept of cryptographic hash function with infinite output and can perform quasi all symmetric cryptographic functions, from hashing to pseudo-random number generation to authenticated encryption.

We use HMAC-SHA-256 for the message authentication as it is stronger for authentication process; AES-256 in CBC-mode for symmetric encryption; and 2048-bit RSASSA-PSS for digital signatures. To implement 2048-bit RSASSA-PSS you have to acquire Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files from Oracle Technology Network and IAIK (Institute for Applied Information Processing and Communication) crypto libraries free version for educational purpose. We choose RSA over DSA for digital signatures because of its faster verification speed. To setup a TLS (Transport Layer Security) connection, we generate self-signed SSL (Secure Socket Layer) certificate using *Keytool* in Java. We implemented Nymble System on our local network. To achieve communication through anonymous networks we use Tor Browser Bundle known as Vidalia with Tor software version 0.2.2.37.

We evaluated our system on a 3.20GHz Intel(R) Core i3 CPU with 2 GB of RAM. The PM, the NM, and the server were implemented using Apache Tomcat 7.0 and user portion was implemented as a Firefox ESR 10.0.5. For each experiment relating to Nymble system performance, we report the average of 10 runs.

VII. EXPERIMENTAL RESULTS

These results show average size of credentials acquired by user, blacklist database update time and average size of updated database.

A notable problem with SHA-1 and SHA-2 is that they both use the same engine, called Merkle-Damgard, [14] to process message text. This means that a successful attack on SHA-1 becomes a potential threat on SHA-2. The fact remains that, while no successful attacks against a full-round SHA-2 have been announced, there is no doubt that attack mechanisms are being developed in private. Hence, in our implementation we use Keccak algorithm as a cryptographic hash function which eliminate the possibility of attach over Keccak.

A linkability window of one day with five minutes time periods equates to L = 288. The size of credential in this case is 88 KB, the size of a blacklist without complaints is roughly 10 KB. Figure shows graphical representation of variations in the size of data structures used.

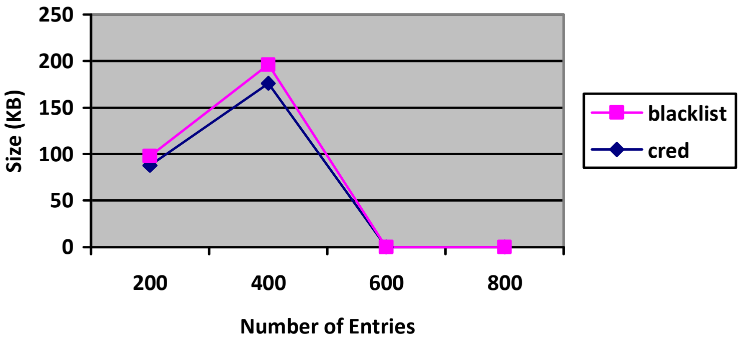


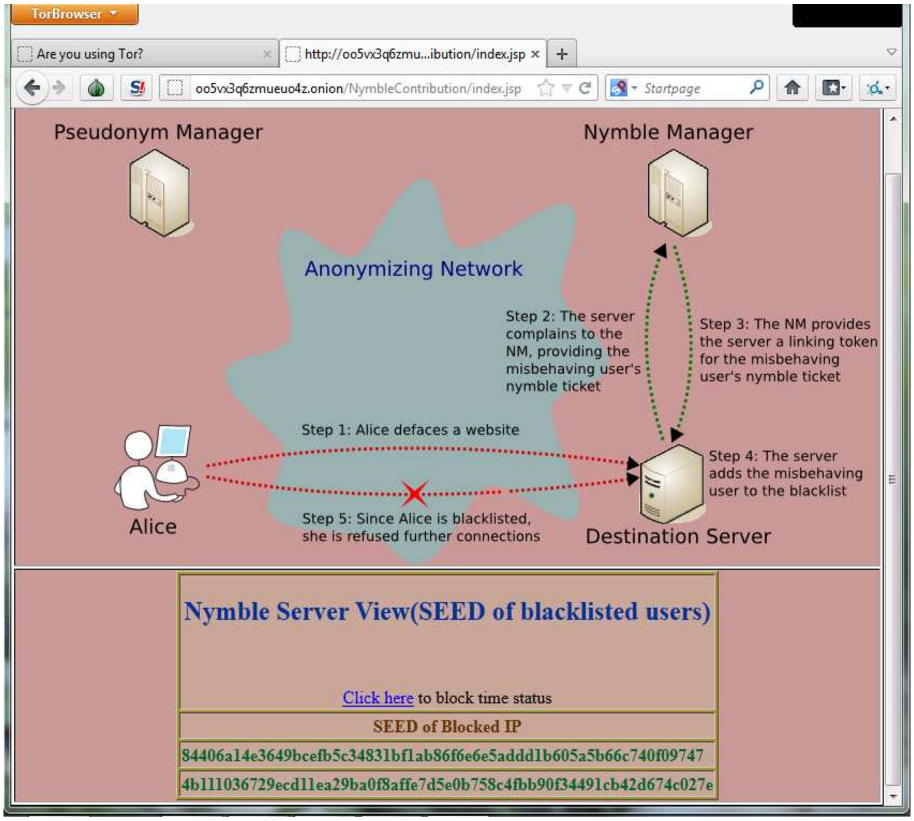
Figure 2. Size of various data structure

The following TABLE II shows time required to execute each module on local network.

TABLE II

Module	Time in millisecond
Nymble Manager	2233
Pseudonym Manager	380
Server	634
Credential creation	195

Following Figure 3 shows the blacklist of users retrieved form database. It shows the seed computed by server for a particular misbehaved user.



Step 2: The server complains to the NM, providing the misbehaving user's nymble ticket

Step 3: The NM provides the server a linking token for the misbehaving user's nymble ticket

Step 1: Alice defaces a website

Step 4: The server adds the misbehaving user to the blacklist

Step 5: Since Alice is blacklisted, she is refused further connections

Nymble Server View(SEED of blacklisted users) Click here to block time status
SEED of Blocked IP
84406a14e3649bcefb5c34831bf1ab86f6e6e5addd1b605a5b66c740f09747
4b111036729ecd11ea29ba0f8affe7d5e0b758c4fbb90f34491cb42d674c027e

Figure 3. Blacklisting User

VIII. CONCLUSIONS

We have proposed and implemented security refinements in Nymble system using combination of various cryptographic algorithms. These algorithms add an additional layer of security to anonymity of users over anonymous networks.

We show how these cryptographic 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 anonymous networks such as Tor, which has thus far been entirely restricted by several services because of users who makes evil use of their anonymity.

REFERENCES

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

[2] Patrick P. Tsang, Apu Kapadia, "Nymble: Blocking Misbehaving Users in Anonymizing Networks," IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 2, March-April 2011.

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

[4] P. C. Johnson, A. Kapadia, P. P. Tsang and S. W. Smith, "Nymble: Anonymous IP-Address Blocking," Proc. Conf. Privacy Enhancing Technologies, Springer, 2007, pp. 113-133.

[5] P. P. Tsang, M. H. Au, A. Kapadia and S. W. Smith, "Blacklistable anonymous credentials: Blocking misbehaving users without TTPs," Proceedings of the 14th ACM conference on Computer and communications security, 2007, pp. 72–81.

[6] A. Lysyanskaya, R. L. Rivest, A. Sahai and S. Wolf, "Pseudonym Systems," Proc. Conf. Selected Areas in Cryptography, Springer, 1999, pp. 184-199.

[7] 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, 2005, pp. 136-153.

[8] Jian Ren, Jie Wu, "Survey on anonymous communications in computer networks," Computer

Communications, vol. 33 issue 4, pp. 420-433, 2010.

[9] D. Chaum, "Untraceable electronic mail return addresses and digital pseudonyms," Communications of the ACM, vol. 24 issue 2, Feb. 1981.

[10] M. Reiter, A. Rubin, "Crowds: anonymity for web transaction," ACM Transactions on Information and System Security (TISSEC), vol. 1 issue 1, pp. 66-92, Nov. 1998.

[11] R. Rivest, A. Shamir, Y. Tauman, "How to leak a secret, in: Advances in Cryptology," ASIACRYPT, Lecture Notes in Computer Science, vol. 2248, Springer, Berlin-Heidelberg, 2001.

[12] Bruce Schneier, Applied Cryptography, John Wiley & Sons, 1994. ISBN 0-471-59756-2.

[13] Bruce Schneier, Schneier on Security, John Wiley & Sons, 2008. ISBN 978-0-470-39535-6.

[14] (2002) The Keccak website. [Online]. Available: http://keccak.noekeon.org/

[15] Imad Fakhri Al-shaikhli, Mohammad A. Alahmad, Khanssaa Munthir, "Hash Function of Finalist SHA-3: Analysis Study," International Journal of Advanced Computer Science and Information Technology (IJACSIT), Vol. 2, No. 2, April 2013, Page: 1-12, ISSN: 2296-1739.