S.K.Alisha* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.4, Issue No.3, April – May 2016, 2907 – 2910.

A Practical Approach for Supervision of Tor Networks

S.K.ALISHA
Sr.Asst.Prof.,
Dept. of CSE,
SVES,
Bhimavaram,
W.G.Dt., A.P,India

P. SURESH VARMA
Professor,
Dept. of CSE,
Adikavi Nannaya University,
Rajahmundry,
A.P, India

Abstract: **While the website administrators cannot blacklist individual Internet protocol address of malicious users, they blacklist complete anonymizing system. On the other hand these measures reduce malicious activity through anonymizing networks at the expense of disallowing anonymous access towards behaving users. In our work we provide widespread credential system known as Nymble which is an effective system. It can be used to include a layer of accountability towards any publicly accepted anonymizing network. Here the servers overcome a potential to blacklist misbehaving users, as a result blocking users without compromising their anonymity. The proposed system makes usage of secured cryptographic hash functions, secured digital signatures, and secured symmetric-key encryption as well as data structures for maintaining efficiency. Our work will enhance the majority approval of anonymizing networks, which has, so far, been totally blocked by quite a lot of services because of users who misuse their anonymity. Our system verifies that users are responsive of their blacklist position before they present a nymble, and disconnect instantly if they are blacklisted.**

Keywords: **Internet protocol, Nymble, Anonymity, Malicious users, Blacklisting, Data structures, Cryptographic functions.**

INTRODUCTION

Anonymizing networks such as Tor route traffic through independent nodes in separate administrative domains to hide a client's IP address. However, some users have misused such networks under the cover of anonymity; users have repeatedly defaced popular websites. Since website administrators cannot blacklist individual malicious users' IP addresses, they blacklist the entire anonymizing network. Such measures eliminate malicious activity through anonymizing networks at the cost of denying anonymous access to behaving users. In other words, a few "bad apples" spoil the fun for all. (This has happened repeatedly with Tor.1).In the literature there are large number of solutions to this problem, each providing accountability some extent.

In the systems of pseudonymous credential user's sign into websites by means of pseudonyms, those are added to a blacklist during misbehaviour of a user [2]. However this method results in pseudonymity for the entire users and weakens the anonymity. These schemes suffer from a common weakness that there is little to motivate or prevent a user from sharing his pseudonyms or credentials with other users.

In Group Signatures [3] servers allow to revoke or cancel a misbehaving user's anonymity by complaining it to a group manager. But the servers must have to query the group manager for each and every authentication and hence this system considerably lacks scalability. There is a constraint that servers can easily find users' IP addresses with the use of Traceable Signature.

Traceable Signatures [4] allow the group manager 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.

Backward unlinkability allows subjective blacklisting. Subjective blacklisting is suitable to the servers like Wikipedia where misbehaviours like questionable edits to a Webpage are difficult to define in mathematical terms. With dynamic accumulators [5], a revocation operation results in a new accumulator and public parameters for the group, and all other existing users' credentials must be updated, and it is thus difficult to manage in practical settings.

Verifier-local revocation (VLR) [6] fixes this shortcoming by requiring the server to perform only local updates during revocation. Unfortunately, VLR requires heavy computation at the server that is linear in the size of the blacklist. It is time consuming and is less secure.

In our work we present an efficient system of Nymble, users acquire an ordered collection of nymbles to connect to websites without additional information and using these collections of nymbles anonymous access to services. Servers can blacklist users by obtaining a seed for a particular nymble, allowing them to link future nymbles from the same user. Therefore Servers can therefore

blacklist anonymous users without the knowledge of their IP.

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 the purpose of exposition. In our system any number of anonymizing networks can rely on the same Nymble system. The System uses the Secure cryptographic hash functions, secure message authentication, secure symmetric-key encryption, secure digital signatures and data structures for efficiency.

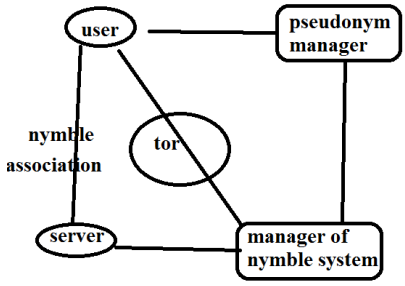

Figure1: Overview of Nymble System.

OVERIEW PROPOSED SYSTEM

The PM knowledge about Tor Routers and users are directly communicating with it as shown in Figure1.Then PM issues pseudonyms to users. A pseudonym pse has two components ny and ma: ny is a pseudo-random mapping of the user's identity (e.g. IP address), the linkability window w for which the pseudonym is valid, and the PM's secret key $PKey_p$; ma is a MC that the NM uses to verify the integrity of the pseudonym. This is described in Algorithm.

Algorithm : Pseudonym

Input: (uid, w) ∈H×M

Output: pnym ∈P

1: Extract $PKey_P$, $NKey_{NP}$ from pmState

2: ny := MC(uid||w, $PKey_P$)

3: ma := MC(nym||w, $macKey_{NP}$)

4: **return** pse := (ny, ma)

After obtaining a pseudonym from the PM, the user connects NM through the Tor network, and requests nymbles for access to a particular server. A user's requests to the NM and then nymbles are generated using the user's pseudonym and the server's identity. These nymbles are thus specific to a particular user-server pair.

The Nymble tickets are bound to specific time periods and the 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). The time periods and linkability windows chronologically are $t_1,t_2…t_k$ and w_1 ,w_2 ,..respectively.

The user connects and misbehaves at a server during time period t' within linkability window w'. The server detects this misbehavior and complains to the NM in time period t_t of the same linkability window w'. As part of the complaint, the server presents the nymble ticket of the misbehaving user and obtains the corresponding seed from the NM.

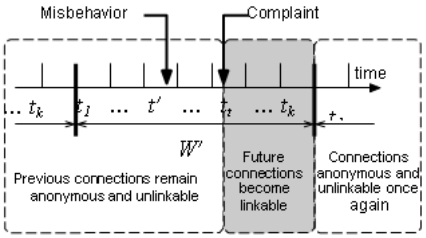

Figure. 2. The life cycle of a misbehaving user

Our system makes sure that users are conscious of their blacklist position before they provide a nymble, and cut off instantly if they are blacklisted. In our proposed system users acquire an efficient collection of nymbles, a particular type of pseudonym, to connect the websites. The system uses Building Blocks and Data Structures for its functioning.

MAJOR BUILDING BLOCKS AND DATA STRUCTURES

The system makes use of building blocks such as: Secure cryptographic hash functions which are functions of one-way and collision-resistant that resemble unsystematic oracles and denoted by H. Secure message authentication consist of key generation and message authentication code MC computation and denoted by M.

Secure symmetric-key encryption consist of key generation, encryption as well as decryption. Secure digital signatures consist of key generation signing as well as verification.

The system uses the data structures for the evaluation of seeds and nymbles. A nymble is a pseudo-random number, which serves as an identifier for a particular time period. The seeds evolve throughout a linkability window using a seed-evolution function f; the seed for the next time period (s_{next}) is computed from the seed for the current time period (s_{cur}) as $s_{next} = f(s_{cur})$.The nymble (n_t) for a time period t is evaluated by applying the nymble-evaluation function g to its corresponding seed (s_t), i.e., (n_t)=g(s_t).

The NM sets s_0 to a pseudo-random mapping of the user's pseudonym pnym, the identity of the server sid, the linkability window w for which the seed is

valid, and the NM's secret key SKeyN. Seeds are therefore specific to user server- window combinations. As a consequence, a seed is useful only for a particular server to link a particular user during a particular linkability window W'.

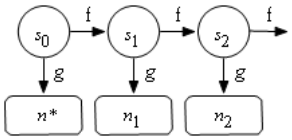

Figure. 3. Evolution of seeds and nymbles. The f and g are two distinct cryptographic hash functions.

PERFORMANCE

To establish a Nymble-connection to a server, a registered 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 credential contains all the nymble tickets for a particular linkability window that a user can present to a particular server. A ticket contains a nymble specific to a server, time period, and linkability window. The ctxt is encrypted data that the NM can use during a complaint that involves the nymble ticket. In particular, ctxt contains the first nymble ($nymble_0$) and the user gets the sequence of nymbles using seed function. During the complaint, the NM can extract the user's seed and issue it to the server by evolving the seed, and $nymble_0$ helps the NM to recognize whether the user has already been blacklisted or not. A server's blacklist is a list of nymbles corresponding to all the nymbles that the server has complained about. Users can quickly check their blacklisting status at a server by checking to see whether their $nymble_0$ appears in the server's blacklist.

A server complains to the NM about a misbehaving user by submitting the user's nymble ticket that is used in the offending connection. The NM returns a seed, from which the server creates a linking token, which contains the seed and the corresponding nymble. Each server maintains a list of linking tokens called the linking-list, and updates each token on the list at every time period. When a user presents a nymble ticket for making a nymble-connection, the server checks the nymble within the ticket against the nymbles in the linking-list entries. A match indicates that the user has been blacklisted. Servers update their blacklists for their purposes; the server needs to provide the user with its blacklist for the current time period during a Nymble-connection establishment and the server needs to be able to blacklist the misbehaving users by processing the newly filed complaints. When there is no complaint blacklists remain unchanged. When there are complaints, the new entries are added to the blacklists. The server updates its blacklist upon its first Nymble-connection establishment request in a time period t.

In Figure 4 The X-axis represents the number of entries which consists of complaints in the blacklist update request, for tickets in the credential (equal to L, the number of time periods in a linkability window w'), nymbles in the blacklist, tokens and seeds in the blacklist update response, and nymbles in the blacklist and the Y-axis represents Size in KB. The tickets in the credential (equal to L, the number of time periods in a linkability window w'), and nymbles in the blacklist. In general, each structure 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 as a complaint list when the server wishes to update its blacklist.

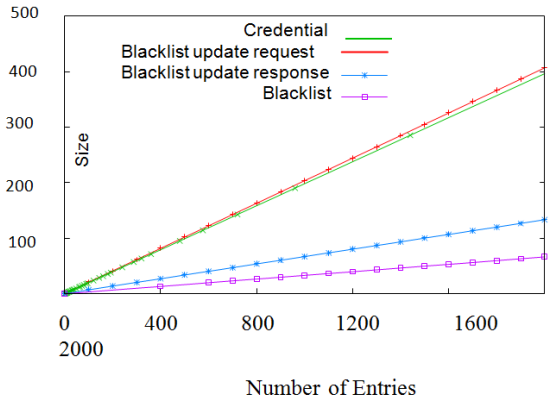

Figure.4. The Performance of Nymble System.

For example, a linkability window of 1 day with 5 minute time periods equates to L = 576.The size of a credential in this case is about 120 KB. The size of a blacklist update request with 100 complaints is 25 KB, whereas the size of a blacklist update response for 100 complaints is only about 8 KB. The size of a blacklist with 1000 nymbles is 34 KB.

CONCLUSION

Here an efficient system of Nymble was presented. Users obtain a well-organized collection of nymbles to connect to websites without extra information and by these collections of nymbles anonymous access to services. In our proposed scheme any number of anonymizing networks can depend on same Nymble system. While our work applies towards anonymizing networks, we make a consideration of Tor for exposition purpose. In our system, the user can download server's blacklist as well as verify their status and when blacklisted, the user disconnects straight away. Our system presents subjective blacklisting; quick authentication speeds, and undetermined authentication, backward unlink ability, and handle sybil attack to make its usage convenient.

REFERENCES

[1] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In CRYPTO, LNCS 1880, pages 255–270. Springer, 2000.

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

[3] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In CRYPTO, LNCS 1880, pages 255–270. Springer, 2000.

[4] Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable Signatures. In EUROCRYPT,LNCS 3027, pages 571–589. Springer, 2004.

[5] Jan Camenisch and Anna Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In CRYPTO, LNCS 2442, pages 61–76. Springer, 2002.

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

