ENHANCING SECURITY BY BROADENING NYMBLE SYSTEM IN ANONYMIZING NETWORKS

Yenumula Sankara Rao*1 and Pradeep2

1Student, M.Tech (CSE), Anadapuram, Al-Meer College of Engineering and Technology, VIZAG, A.P, India.
sankararaoyenumula@gmail.com
2Asst.Professor, (CSE), Anadapuram, Al-Meer College of Engineering and Technology, VIZAG, A.P, India

Abstract - This paper presents Nymble, a system in which servers can blacklist misbehaving users, thereby blocking users without compromising their anonymity. Our system is thus agnostic to different servers’ definitions of misbehavior servers can blacklist users for whatever reason, and the privacy of blacklisted users is maintained. In pseudonymous credential systems users log into Web sites using pseudonyms, which can be added to a blacklist if a user misbehaves. Anonymous credential systems employ group signatures. Basic group signatures allow servers to revoke a misbehaving user’s anonymity by complaining to a group manager. Anonymous networks allow anyone to visit the public areas of the network. Here users access the Internet services through series of routers. This hides the user’s identities and IP address from the server. This may be an advantage for the misbehaving users to destroy popular websites. To avoid such activities, servers may try to block the misbehaving user. But it is not possible in case of anonymous networks. Since the servers are not aware of the user’s identities or IP address it is not possible to block the particular misbehaving user. In such cases servers may block the entire network which also affects normal behaving users. To overcome this problem, a system is designed in which servers can blacklist the misbehaving users without compromising their anonymity. The proposed system accurately finds the misbehaving users and also maintains the blacklist users details in the server. In this system servers can blacklist users for any reason and also the privacy of blacklisted users is maintained.

Keywords— Anonymous blacklisting, privacy, revocation, Pseudonym Manager

INTRODUCTION

An anonymous network is the network that provides anonymity and privacy to the users. Usually user gets connected to the target server with user’s identities, IP address or sharing other computer information. This information can be used to track the user. However, a certain degree of anonymity can be achieved by using a proxy server or a VPN server[1]. But in anonymous networks some information are hidden such that what to be hidden and from whom to be hidden. First we discuss about what to be hidden. Sender anonymity – attacker should not be able to identify the sender of a particular message. Receiver anonymity – attacker should not be able to identify the receiver of a particular message. Unlinkability – though the attacker determines the sender and receiver, he/she should not be able to determine the association. i.e. who communicates with whom. second is from whom to be hidden. Communication partner, External attackers and internal attackers. Crowds [2] and Tor [3] (Third party Onion Routing) are some kinds of anonymous network in which anonymity can be achieved by using a series of routers such that each router works by redirecting the communication through itself. The user’s IP address is then only shared with the entry router while the target server only sees the exit router in the network. Fig. 1, shows how a tor network performs multilayered encryption through a series of routers.

Thus anonymous networks work by placing a wall between the user and the server he/she is visiting. These networks can be used to bypass the restrictions and visit sites that might be blocked in specific country, office or institution. Some people only use these servers to protect their online identity. Anonymous networks are generally useful to internet users who want to ensure that their sessions or activities cannot be monitored. For instance, it is used to circumvent traffic monitoring by organization which want to find out or control which websites employees visit.

Anonymizing networks such as Tor route traffic through independent nodes in separate administrative domains to hide a client’s IP address. Unfortunately, some users have misused such networks—under the cover of Anonymity, users have repeatedly defaced popular Web sites 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 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 Web sites 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 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 unlink ability that we desire, where a user’s accesses before the complaint remain anonymous. Backward unlink ability 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 unlink ability need to pay careful attention to when and why a user must have all their connections linked, and users must worry about whether their behaviors will be judged fairly.

Scope Of The Concept:
The success of such networks, however, has been limited by users employing this anonymity for Abusive purposes such as defacing popular Web sites. Web site administrators routinely rely on IP-address blocking for disabling access to misbehaving users, but blocking IP addresses is not practical if the abuser routes through an anonymizing network. As a result, administrators block all known exit nodes of anonymizing networks, denying anonymous access to misbehaving and behaving users alike.

RELATED WORK

Various Issues in the Existing Approaches:
There are many solutions for the problems and difficulties in anonymous networks. But each method has some limitations and issues. They are as follows: In pseudonym Systems, an individual will be known to other users by a pseudonym which is blacklisted if a user misbehaves. But this results in pseudonymity for all users and weakens the anonymity. Also the users should be prevented from sharing their pseudonyms. Group signature is a method by which a member of a group anonymously signs the message on behalf of the group. Hers the server sends complaints to the Group Manager (GM) if a user misbehaves which lacks scalability. Traceable signatures traces the signatures signed by a single party without opening the signature and revealing the identities of another users. This method does not provide backward unlikability, where the previously issued signatures remain anonymous even after the signer’s revocation. Since there is no backwardunlikability, there will be no subjective blacklisting. Subjective blacklisting is the process by which the server can blacklist the user for whatever reason the server desires. Dynamic accumulator is cryptographic accumulator that allows dynamically adding or deleting a value. But here a single revocation operation results in a new accumulator and public parameters for the group. Thus Updating all the values is impractical. In Verifier Local Revocation (VLR), the verifier performs local updates but there will be heavy computation at the server or the verifier. These approaches do not provide revocation auditability by which the users can verify their status before accessing the server.

ARCHITECTURE OF PROPOSED SYSTEM

The proposed system for blocking misbehaving users involves some major steps. They are as follows: User registers with Pseudonym Manager (PM) by sending user id and IP address PM generates pseudonym and sends back to the user. User sends the received pseudonym and server id to the Security Manager (SM) SM verifies the pseudonym using a key shared between the SM and PM If the pseudonym is valid, SM sends credentials to the user before accessing the server, user can verify whether he/she is blacklisted or not If the user is not blacklisted, connection is established and the user starts accessing the server. If misbehave found server sends complaints to the SM. SM verifies and updates the blacklist database Server disconnects the misbehaving user and blocks the user from further access to the server.

A system having properties like anonymous authentication, backward unlink ability, subjective blacklisting, and fast authentication speeds, rate limited anonymous connections and revocation auditability is introduced. These properties can be implemented by introducing two trusted third parties namely Pseudonym Manager and Blacklist Manager. The system architecture consists of user, service provider and trusted third parties the blacklist manager and the pseudonym manager. If the user wants to execute web transaction, the user first registers with the Pseudonym Manager and issues user with pseudonym based on the IP address provided by it. The service provider registers with the Blacklist Manager which issues set of unique set of tokens. The user using its pseudonym name access the service provider through the anonymizing network. The service provider transfers the pseudonym to the Blacklist Manager. The Blacklist Manager consists of blacklist table. It has attributes like pseudonym, unique token, and blacklist value. Before the Service Provider give access, it checks with the blacklist table, if the pseudonym is present in the blacklist table, then the user is denied access to SP, else the user can access freely through the network and the Service Provider.

All the connections of the user before the control was made
will be unlinked i.e., the accessing information of user can’t be traced back. But after the complaint is made, were all the connections of the user will be linked. This above property helps in maintaining anonymity of the user, even though it is blacklisted.

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

**Blacklist ability:**
This property assures that any honest server can block misbehaving users. If an honest server complains that a user misbehaved in the current time period, then the complaint will be successful and the user will not be able to establish a connection to the server successfully for the following time periods.

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

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

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

**Module Description:**

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

**User Registration:**
A user with identity uid must register with the PM once in each likability window. To do so, the user initiates a type-Basic channel to the PM, followed by the User Registration protocol described below. A login generally requires the user to enter two pieces of information, first a user name and then a password. This information is entered into a login window on a GUI (graphical user interface)[4]. A user name, also referred to as an account name, is a string (i.e., sequence of characters) that uniquely identifies a user. User names can be the same as or related to the real names of users, or they can be completely arbitrary. A password is likewise a string, but it differs from a user name in that it is intended to be kept a secret that is known only to its use.

**Message authentication code (MAC):**
Secure cryptographic hash functions. These are one-way and collision-resistant functions that resemble random oracles [5]. Denote the range of the hash functions by H. Secure message authentication (SMA) [6]. These consist of the key generation (MA.KeyGen), and the message authentication code (MAC) computation (MA.Mac) algorithms. Denote the domain of MACs by M.Secure symmetric-key encryption (Enc) [7]. These consist of the key generation (Enc.KeyGen), encryption (Enc.Encrypt), and decryption (Enc.Decrypt) algorithms. Denote the domain of cipher texts Secure digital signatures (Sig). These consist of the key generation (Sig.KeyGen), signing (Sig.Sign), and verification (Sig.Verify) algorithms. Denote the domain of signatures.

**Pseudonym Manager:**
The user must first contact the Pseudonym Manager (PM) [8] and demonstrate control over a resource; for IP-address blocking, the user must connect to the PM directly. We assume the PM has knowledge about Tor routers, and can ensure that users are communicating with it directly. Pseudonyms are deterministically chosen based on the controlled resource, ensuring that the same pseudonyms always issued for the same resource.

**Nymble Manager:**
After obtaining a pseudonym from the PM, the user connects to the Nymble Manager (NM) [9] through the anonymizing network, and requests nymbles for access to a particular server (such as Wikipedia). A user’s requests to the NM are therefore pseudonymous, and nymbles are generated using the user’s pseudonym and the server’s identity. These nymbles are thus specific to a particular user-server pair.

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

**METHODOLOGY**
The main entities of this framework are the Pseudonym Manager, Blacklist Manager, user and the Service Provider. The Pseudonym Manager does the registration of the new user, authentication and verification. The blacklist manager issues the unique tokens, maintains a blacklist table and link/unlink option based on whether the user access. The
Service Provider registers with the blacklist manager which maintains a blacklist table.

**Blacklist Manager and Pseudonym Manager:**
Blacklist manager controls the entire process of the whole architecture. Both the Service Provider and the Pseudonym Manager can be only accessed through the Blacklist Manager. It contains blacklist table which contains the attributes of pseudonym, tokens, blacklist status. Pseudonym manager controls the user activities. Users can access the anonymizing network only if they register with the Pseudonym Manager. It has knowledge about the routers in the anonymizing network. Pseudonyms are chosen based on controlled resources such that no two users can have the same pseudonym. The user connections are anonymous to Pseudonym Manager. It is created to reduce the load from the Blacklist Manager. It also acts as second server to the system.

**Time:**
Tokens generated by the blacklist manager are bound to specific time periods called the likability window. This likability window is again divided small time intervals. Users’ access within a time period is tied to single token generated by the Blacklist Manager [10]. The use of different tokens across time periods grants the user anonymity between time periods — smaller time periods provide users with higher rates of anonymous authentication, and likewise longer time periods rate-limit the number of misbehaviors from a particular user before he or she is blocked.

**Notification of Blacklist Status:**
Users expect their connections to be anonymous while using the anonymizing network. In case of misbehavior of user, its future connection will be linked. Therefore the user should be able to view their blacklist status while trying to connect to the service provider. The user is able to download the blacklist table can check whether he is on the list. If present, user disconnects immediately.

**User Details and User Access Control:**
All the user details and the access history will be maintained at database of both the pseudonym manager and the service provider. If the user misbehaves all those history will be updated to both the database.

**ANALYSIS AND RESULT**

Fig 4 shows size of the entities used in this framework. The x-axis shows the number of entries i.e., the complaints in the blacklist update request, tokens generated by the blacklist manager [10], seeds in the blacklist update response. Assume L be the number of time periods in the linkability window. And credential is the collection of tokens.

If likability window of one day is 5 minutes, then time period L=288. Each entity grows as number of entities grows. Credential and blacklist update request grows with the same rate because credential is same as complaint list sent when the blacklist table is to be updated.

The likability window has two purposes i.e. allows dynamism since resources like IP address can get reassigned among different users so that is difficult to blacklist the resource for long time and it ensures forgiveness for some time after being misbehaved.

**Permission Control and Blacklist:**
All the user details are maintained both at Pseudonym Manager and the Blacklist Manager. Authorized and unblocked users can access the anonymizing network. Blocked user can’t access the network, but with the permission of the Blacklist Manager it can access. If the user misbehaves, service provider will link the future connections within current linkability window. Even though misbehaving user can be blocked from making any other connection, users past connection remain unlikeable. The service providers can subjectively judge users for any reason, since the privacy of users are maintained.

![Figure 3. Linkability Window](image)

![Figure 4. Size of entities Vs Number of entries](image)

![Figure 5. Performance Vs Number of entries](image)
that occurs only once every link ability window for each user wanting to connect to the service provider. For blacklist updates, the initial jump in the graph corresponds to the fixed overhead associated with updating the blacklist. If there is no complaint then it takes blacklist to less than millisecond for updating.

FUTURE WORK

We hope that our work will increase the mainstream acceptance of anonymizing networks such as Tor, which has been completely blocked by several services because of users who bussed their anonymity.

Advantages:

a. 1. Our system ensures that users are aware of their blacklist status before they present a nymble, and disconnect immediately if they are blacklisted.

b. 2. In our system, the user can download the server’s blacklist and verify her status. If blacklisted, the user disconnects immediately.

CONCLUSION

A new system is proposed that adds an additional layer of security to the anonymous networks. This system is used to block the misbehaving users in anonymizing networks. It is automatically finds the misbehaving user and blacklists them without affecting their privacy and anonymity. This method is a cryptographic construction that provides anonymous authentication, fast authentication speeds, subjective blacklisting, and backward anonymity and revocation audit ability. This method is practical, effective and efficient to the needs of both users and services. The proposed method motivates the need for security in anonymous networks and this system will increase the acceptance of anonymous networks that is blocked by several services because of users who misuse their anonymity. Currently the proposed system has been simulated with PM, SM and server on the local network. In future this work will be enhanced to work on a remote machine. This work can also be extended into a multiple rounds of pseudonym construction in which the PM participates in multiple rounds of communication with the user. This adds one more layer of security to the system. We have proposed and built a comprehensive credential system called Nymble, which can be used to add a layer of accountability to any publicly known anonymizing network. Servers can blacklist misbehaving users while maintaining their privacy, and we show how these properties can be attained in a way that is practical, efficient, and sensitive to the needs of both users and services. We hope that our work will increase the mainstream acceptance of anonymizing networks such as Tor, which has, thus far, been completely blocked by several services because of users who abuse their anonymity.

REFERENCES


