Network Anonymity

Sardar Muhammad Sulaman, Muhammad Roohan Kebria
Email: {sarmu544, muhke224}@student.liu.se
Supervisor: David Byers {david.byers@liu.se}
Information Security Second Course
Linkoping University, Sweden

Abstract

In this paper we will explore the concept of anonymity and its importance, the available technologies and the degree of anonymity they provide to a user. Why one thinks it is important to use anonymity in their daily life and others want to defeat it. Our focus will be on different implemented anonymous systems, how they work and what they provide. We will analyze these implemented anonymous systems based on the inherit latency they have. We will mainly focus these systems under the division of low-latency and high-latency anonymous system, and the vulnerabilities inherit in these anonymous systems.

1. Introduction

Anonymity is a Greek word, meanings namelessness or without any recognizable identity. Anonymity is a result of some action in which a person intentionally hide her identifiable information characteristics [1]. Hiding information can be for good or bad purpose. An entity in a system is said to be anonymous if no other can identify the first entity and there is no way to correlate actions to a specific entity [2]. Another term weakly related to anonymity is pseudonymity, in which one can identify the subject somehow but it is unable to correlate the actions to the same entity.

By design there is no privacy across the internet and the applications used for a certain purpose. Web browsers present internet address, domain name, platform, organization and cookies to the contacted parties. The request by the client always logged at the server which reveals where the client resides exactly and the other characteristics. There is neither privacy nor anonymity. This way a user can be vulnerable to certain attacks. The attacker might be the target machine or an eavesdropper in between client/server who can actively launch an attack to gain information that might not be revealed to a common user, so presenting a thread against the users undermining their confidentiality, integrity and availability. Anonymity on the internet is related to certain aspects in modern days including e-commerce, data communication, VPN, NAT, web browsing, relay chat, remote login etc.

There are certain levels of anonymity protection covering a whole communication pattern includes sender anonymity, receiver anonymity, unlink-ability, information anonymity and as a whole network anonymity. This property is used commonly when someone interested in correlating certain actions with a particular entity through assumptions only. Hence anonymity provides reduced accountability and preserves one’s reputation.

2. Background

Being anonymous and use of different methods to remove traces is not a new term. Looking at the history we noticed that there are certain stories and myths related to an anonymous person and still believed to be correct. For long time people use to hide sensitive information with the help of mathematical functions, making it difficult for a common person to extract the exact meaning.

There are many concepts with different techniques and these are used with real applications in order to provide certain level of anonymity to their users. One of these worth mentioning includes “Russian Dolls”, a set of dolls (varying size) placed one inside another and no one exactly knows the number of dolls inside. This provides the basis for modern “Onion Routing” [1], which we will discuss later. Another conceptual approach in modern era keeping internet anonymity in mind is the invention of “Mixnet” [8] by David Chaum. David Chaum was the first person who introduced the idea of anonymity for internet use. He introduced an anonymous system for emails in 1981 [5]. The basic technique was to use a special mail server known as Mix for the retrieval and delivery of emails. Mix was a computer placed in between the sender and the receiver to provide anonymous communication by means of cryptography (public-private key). Chaum’s Mixes provide sender’s anonymity and protection against traffic analysis.

This was just the start of anonymous systems for digital communication, now days we have several implemented anonymous systems used differently for different purposes. There are certain aspects for the division of available anonymous systems [2, 8], but we have chosen latency for analysis.

- Low-Latency Anonymous Systems
- High-Latency Anonymous Systems

Latency is amount of time experienced while processing and forwarding an incoming packet to a specific interface.

3. Solutions and Analysis

In this section we will analyze available solutions for anonymous communication by looking into respective implemented technologies and adopted work strategy.

3.1. Low-Latency Anonymous Systems

Implementing the basic concept introduced by David Chaum encompasses large delay during communication, which is tolerable for the email, file transfer etc. The real time interactive applications (web browsing, voice, video, and gaming) are unable to tolerate long delay. So latency needs to be reduced to certain extent making a system feasible for a broad range of applications while providing sufficient anonymity. Low latency anonymous systems answer this. To understand this idea a brief description of such systems is provided.

3.1.1 Anonymous Proxies

The anonymous proxies can be used by those users who don’t have much information about the anonymity. These users can secure their internet activities by using freely available anonymous proxies. The web anonymous proxies work in client-server mode, making them suitable for HTTP traffic. The client sends a request for a URL to the anonymous proxy, which sends request to mentioned web server and on getting response returns it back to the client. The requested web page seems to come from some other address than expected. By using these proxies a user can bypass the local restriction. The anonymous proxies also change the URL field to provide confrontation by hiding ones identity and activity.

These anonymous proxies also have some vulnerability despite easy use and access. A proxy has full control over the user information which is not good for application having credentials like e-business. A proxy can actually see what is going through, it can be a bottle neck and open as being a single point of failure regarding security. To get a better service it is possible to do proxy chaining at the expense of complexity and latency, resulting in network architecture (like TOR that we will discuss later). Also there is no encryption provided by most of the proxies hence making it feasible to trace the contents. In short, using a proxy to hide information is not a solution at its own but an excuse which is better than being clear to an attacker.

3.1.2 Onion Routing

Onion Routing is used practically in real-time, which provides an application independent anonymous communication over a public network. This prevents the communication being tracked independent [2] of application used. It is build upon the concept of Russian Dolls by encrypting a message several times and sending across the Onion network that consist several Onion Routers (OR). An OR is like an ordinary router but having “MIX” functionality [2].

The concept is built upon “Onion”, which is a layered structure. An onion like layered data structure has to be exchanged in between onion routers. The path taken by the onion message is decided at the ingress OR which encrypts the clear message in layered structure with the public key [2] of the onion routers between the source and destination. The onion message also has the next hop information and key seed material. The seed is needed for generation of symmetric key [2]. This symmetric key is used to decrypt a layer of the incoming encrypted onion message and the next hop information is used to forward the onion message with remaining encrypted layers to the next hop. Each Onion router removes a layer of encryption for retrieval of routing information and this process is executed repeatedly at each OR until the message arrived at the destination unencrypted. This layered encryption prevents from information revelation about sender, receiver and contents of the message to the intermediate nodes [5]. This is not an end to end solution for anonymity, because traffic from sender/receiver to an Onion router is not encrypted.

The sender application has to connect with a particular OR via an application proxy and onion proxy. As the sender wants a connection to a destination an anonymous connection setup is required [2]. The first OR has all the information that will be used to transmit a message from source to destination. The application proxy first anonymizes the data coming from the sender and then encrypts it several times using the symmetric keys distributed among the ORs on the path [2]. The OR is resilient to traffic analysis, because there is no correlation between different parts of a message. Each onion message passes through different OR from source to destination making it impractical for a person to deduce which packet is connected with which stream.

3.1.2.1 TOR

TOR “The Onion Routing” project [6] is freeware open source software implements the onion routing mechanism that aims to hide user identity and actions from being monitored.

It can work with many existing applications like web browsers, instant messaging clients and other applications based on TCP protocol. It has client software and a large number of servers that hide the personal information of the user. The users of TOR system must have an onion proxy presenting a SOCKS interface on their side. TOR creates a virtual circuit consisting of onion routers, using multi layer encryption and concealing the identity of user. As the message travel through the TOR network, each TOR router decrypts its own layer and retrieves the forwarding information for routing to next hop. TOR is an application independent and works at the transmission control protocol TCP stream level [7]. The SOCKS is an internet protocol that routes an IP packet in client-server paradigm using proxy in between.

Data packets over the internet are divided into two parts, data payload and the header. The data payload carries the data being sent and the header contains the information required for the routing. If data is sent in encrypted form still it can reveal a great deal of information about ones activities, because of the unencrypted header which provides information e.g. source address, destination address, size etc. In Simple type of traffic analysis attack an attacker sits between sender and receiver and performs the traffic analysis. But some powerful analysis attacks in which an attacker does not need to sit between the sender and receiver; he/she can monitor the traffic remotely. Encryption cannot solve this problem, because useful information can be revealed from the header, which has enough information for the attack. TOR prevents to some extent from happening of this simple and modern traffic analysis attacks. It does by having an indirect connection between source and destination. Data sent over the TOR uses a random path instead of a single path through several relays that cover the user’s track; no one can find the path by observation. The basic idea is that one relay cannot see more than one hop in the network which prevents the tracking of route of the messages [7].

TOR is the well known implemented anonymous system at the moment; it provides better security, efficiency and implementation by introducing some improvements in the original onion routing. The first improvement is about the clients that how they will learn existing ORs in the network, the onion routing does not provide any specification regarding this. The TOR design solved the above problem by using a directory server. This directory server has the duty of collection of the information about the available onion routers in the network. Each onion router that joins the network for first time it gets the information about existing TOR routers from the directory server. This information is also cached on several servers for load balancing, as the central directory server can becomes the performance bottleneck and single point of failure.

The next improvement for TOR design in onion routing is about the connection establishment by the client. The initiator of the connection exchanges the transient session keys with every router in its path by using the Diffie-Hellman key exchange algorithm and RSA authentication [8]. After creating and exchanging the session key with the first router, client repeat the same process with the next hop using encrypted tunnel until it reaches at the destination router while completing the circuit. TOR provides "perfect forward secrecy" [8] and prevention from replay attacks by discarding the transient session keys at the circuit teardown.

The last improvement in the original onion routing for the TOR implementation is known as "location-hidden services" [8]. These services provide anonymity to the web servers for the TOR clients by running some special server software. Everyone can access these servers but does not know the IP address and residual location. Location-hidden services maintain the anonymous connection to the small part of the systems available in the network; this small part is known as "introduction points" [8]. The client connects with the hidden services through the introduction point and refer to another system known as "rendezvous point" [8] by using Diffie-Hellman key exchange algorithm. As both the client and the hidden-services connects to the rendezvous point for the creation of the encrypted tunnel, TOR introduce the "entry guards" [8], small set of the nodes which gives choice for the selection of first hop during circuit establishment. The entry guard prevents the certain type of attack which reveals the information about the hidden-services. So, the TOR is a good solution for anonymity over the internet.

3.1.3 Java Anon Proxy (JAP)

JAP is a proxy system that provides anonymous web browsing. It is a proxy server that has some integrated services and can be used with any web browser. It enables the web browser to provide the anonymous web browsing by hiding the original IP address. JAP uses single static IP address instead of real IP address and shares this among many JAP users. Having this shared static address the visited sites and an attacker who is doing traffic analysis are unable to detect information about the source communicating over the internet.

JAP client does not directly connect to the web server; it connects to the several intermediate "Mix" [2] systems using encryption. It uses prearranged list of the mixes, which provides a further list of interconnected mixes known as "Mix Cascade". Every Mix Cascade consists of three Mix systems. The JAP client first connects and registers with the infoService system, this system determines whether the JAP client is still compatible with Mix software or not. After this JAP client connects and registers with the first Mix system of the selected mix cascade. Each mix system of Mix Cascade cryptographically transforms and reorders the data (request) before forwarding it to the next Mix system [2]. The JAP client after encryption sends the data to the first registered Mix system. The Mix system mixes the received data with the data (requests) of other users and sends it to the next Mix system. Next Mix system performs the same task like first. After mixing it forwards the data to the third Mix system which is the last Mix station in the Mix Cascade. Third Mix system forwards the anonymized web request to the internet via the "cache proxy" [2].

JAP has a restriction for the prevention of traffic analysis attack. The restriction is, the JAP client has to send the data to the first Mix of selected Mix Cascade in a constant rate. If client does not have any data to send, then it has to send the dummy data. This is also suggested between the last Mix of cascade and the cache proxy. Above restriction is theoretical, but the implemented systems don't use any dummy traffic due to excessive load over the network.

3.1.4 Crowds

It provides a mechanism for anonymous web browsing. With Crowds at the client side, the web server is unable to determine the credentials of a client's request that could be uniquely identified. It operates by splitting the users into groups across geographical area.

After creating groups it sends the web request of a user on behalf of these groups while hiding the identity of that user.

The main idea is to blend a user among the crowds of computers or users. In Crowds a user is represented as a process known as Jondo [8]. Jondos are assigned to a crowd by an administrative process known as Blender [8]. When the client's web browser makes a request, his/her jondo does a random selection of another jondo from the Crowd to create a random path. The selected jondo flip the coin [8] while having forwarding probability greater than 0.5. Depending on the outcome of the flip it decide among, whether it has to select another jondo from the crowd to forward the request or simply forward the request by itself to the web server. All jondos keep track of the path taken by the request so that upon receiving reply it correctly maps it to the requesting jondos. The connection between the pairs of jondos is encrypted by the symmetric key provided by the blender to the jondo when it enters the crowd. This is loosely related to the onion routing. So each jondo can forward the request for itself and also for the other jondos, the receiving jondo cannot distinguish about who is the originator of this request [7].

The crowd hides the user's communication details by routing them among the group of similar users. If we have crowd protocol than a local eavesdropper or a corrupt group member by doing traffic analysis cannot be sure about the sender of particular traffic or message [8].

3.1.5 Tarzan

Tarzan is a low latency anonymous system having very close relation with onion routing. Like onion routing the initiators in the Tarzan creates circuit in the network by generating shared keys for each hop and repeatedly encrypting it with the public keys of the server in the network [8]. These shared keys used for forwarding the data over the established circuit. Unlike Tor, Tarzan uses UDP as transport protocol.

In Tarzan every member can forward traffic for other same like Crowds. Tarzan uses a peer-to-peer "gossip" protocol for collection of information about the other servers in the network. When a new node initializes for the first time it asks for available servers in the network from its neighbor selecting it randomly. Then the selected neighbor asks the same question from its own neighbor again selecting randomly and this process continuous and collects information about the all available servers in the network [8]. It also prevents the link creation by the global passive adversaries. Whenever a new node joins the network, it chooses N other nodes as *mimics, and* asks them to exchange information as *mimic traffic* (Cover traffic) with it. Each node selects mimics by the repeatedly use of cryptographic hash functions on the IP address and the current date. The mimic selections are also symmetric; if it is not symmetric then nodes can only send data on their outgoing link.

The initiator of the anonymous network creates a circuit through Tarzan network by choosing first hop randomly from its set of mimics, and the second hop is chosen by the first hop from its set of mimics and this process goes on creating the circuit. The first hop in the circuit cannot distinguish about traffic it is receiving as either cover traffic or real traffic. Finally, traffic is forwarded over the network using layered encryption same like the onion routing. Due to "cover traffic" Tarzan provides resistance from analysis.

3.2 High Latency Anonymous Systems

High latency anonymous systems provide strong anonymity as compared to the low latency anonymous systems. We cannot tolerate long delays in interactive applications; due to such restriction we cannot use high latency anonymous systems for the interactive application. High latency systems are implemented for the non interactive applications and provide higher degree of anonymity. Now we are going to discuss some of the implemented high latency anonymous systems.

3.2.1 Garlic Routing

The Garlic Routing [9] is a variant for the Onion Routing. The Garlic Routing encrypts multiple messages together into one single onion like message. It increases the degree of anonymity by multiplexing and routing the messages along the path towards their destinations. Hence providing better protection against the traffic analysis attack and eavesdropping.

The onion multiplex message is known as “Cloves” [9] in garlic routing. It provides additional options to be used at certain hop along the path during transmission. These options include the introduction of a certain delay at the next node specifying how much time this incoming packet can wait without disturbing an ongoing communication. The mentioned delay can be more than the traditional encrypted packet delay. These options also specify where to decrypt the clove, which provides multiple encrypted onions for the delivery to their destinations. It also provides padding option which prevents the information disclosure about the number of onion messages in the clove. By implementing the garlic routing we can achieve end-to-end anonymity but it falls in high latency anonymous systems because of explicitly specified delay and computation time to encrypt and decrypt multiple messages.

3.2.2 Remailers

Emails also reveal the identity of the receiver and the sender by the “To and From” fields. These fields exist in all emails sent across the network and can be used for eavesdropping. For achieving anonymity at this level one can make a pseudonymous email account for his certain job function. But often this is not suitable in a situation when someone is working in a group and bound to use a specific email account for the job responsibilities. This restriction introduces the need of anonymous communication for emails.

There are certain ways to provide anonymous communication using “Remailers”. As the name depicts their main functionality is to accept an incoming mail, store it, process it with certain requirements and pass it to the destination. We have certain types of remailers providing different degree of anonymity for the email.

Type 0 Remailer categorized as pseudonymous server, which just strips the source identifying information from the email and forward to the other end. Type 0 remailer also provides the IP anonymity. This is presented with the introduction of *anon.penet.fi* to the general public [2]. But the email is vulnerable from/to sender/receiver to/from Remailer.

Type 1 Remailer used for achieving the High level of anonymity at the cost of latency. The “cipher chunk remailer” is used for this purpose [2]. In this system an email is composed of encrypted parts of an email (providing an onion like structure as discussed). This encrypted email is sent across a path of mixes. A mix decrypts the message and maps the original source email address with a pseudonymous email address. This mapping is stored for the future use. Mix forwards an email after getting N (pre-defined number) emails within an allowed delay. This mechanism is vulnerable to the spam attack; a mix can be flood by an attacker with number of emails [2].

Type 2 Remailer is used when an attacker gets an email address of the contacted person. For the protection of the user in this situation “Mixmaster” has to be used [2]. The email has been encrypted several times with different symmetric keys. Every mix across the path strips off the top header and gets the information regarding next hop, adds some garbage information to preserve size and forward on the outgoing link. After delay and reordering the message is sent to the next mix, providing anonymous communication.

3.3. Analysis

The systems discussed in this paper provide resilience towards the traffic analysis and eavesdropping through anonymous mechanisms. These mechanisms can be used in flexible manner and for a range of applications as VPN, anonymous chatting, browsing, conversation, cash transfer and remote login. Providing such means reduce the burden from the application developers to provide the requested anonymity (security) functions in the application layer. This makes an application more efficient to be implemented in heterogeneous environment. Diverse application can be used for any type of required anonymous functionality. With the help of these commercially available anonymous systems a normal user can perform work without thinking, whether this application has such functionality inherit or not. Most of these anonymous systems have specific requirements. Everyone can access them free or at lowest possible cost. A single anonymous system is unable to provide all functionalities needed for being completely anonymous.

To achieve anonymity there is some tradeoff to pay for additional functionality. Latency required and degree of anonymity achieved is an example for this. High anonymity requirement require more processing for cryptographic computations that increases latency in communication. As one tool gets popular in certain domain, it became more flexible to hide a certain communication and attracts more users. Having large number of users is good for providing anonymity; it is more difficult to point out a specific subject from a group of users for a specific action. With the large number of users we can achieve higher degree of anonymity.

4. Related Work

Anonymity is a hot topic in today's network environment as most of the jobs are to be done without being traced and people are curious to know about these things. To provide adequate knowledge in general and in specific lots of information about the proposed models and implemented models has been published on different forums around the globe. People can access them, debate on them and can enhance the functionalities to be used for their needs.

Michael G. [1] provides good in-depth knowledge about the needs, threat models and applications of anonymous systems regarding the affects of such systems in social environment. Claessens J. [2] provides the comparison of the available implemented anonymous networks and how they work. The websites [6] of the particular implemented mechanisms provide information about these mechanisms and their practical deployment. Survey reports [8] provide an up to date achievements in this field.

5. Conclusion

Today's world is a digital world, we perform approximately all the daily life activities digitally over the Internet. We use Internet for the web browsing, newspaper reading, online shopping, communication, transfer of data, online banking and many more. While performing such activities the user has a severe threat regarding privacy and personal information. Now a day's data storage equipments are very cheap and one can store such personal information about the users available on the Internet. To prevent this threat we have very nice thing which is known as Anonymous Network Services. Those who want protection from curious attackers must use high latency anonymous systems. On the other hand to use interactive applications one must use low latency anonymous systems.

Network Anonymous Services is a vast area of research and development. Anonymity research field is still young and it does not provide many options for the measurement and comparison of the protection which we get in different situations. We can estimate the interest of general public for anonymity by looking at the available and continued projects for anonymous communication. The anonymity is the demand of general public for secure web browsing and communication over an unsecured network. Most of the anonymous network service projects developed by the academic institutes and by the individuals. After discussing above all systems for network anonymity we came to this conclusion that the Global Network in the future must have privacy and Anonymity as its essential property. For achievement of the above statement, the protocol designers must consider anonymity while developing such products.

References

[1] “Anonymous Connections and Onion Routing", Michael G. Reed, Paul F. Syverson, David M. Goldschlag, IEEE, 1998

[2] “Solutions for anonymous communication on the Internet”, Claessens J., Preneel B., Vandewalle J., IEEE, 1999

[3] “Technical challenges of network anonymity”, D. Kesdogana, C. Palmer, Computer Communications, 2006

[4] “Network Flow Watermarking Attack on Low-Latency Anonymous Communication

Systems", Xinyuan W., Shiping C., Sushil J., IEEE, 2007.

[5] "Locating Hidden Servers" Øverlier, Lasse; Paul Syverson, IEEE Symposium on Security and Privacy, Oakland, CA, 2006

[6] https://www.torproject.org/ - accessed 18 March, 2011.

[7] "Crowds: Anonymity for Web Transactions", Michael Reiter, Aviel Rubin, ACM Transactions on Information and System Security,2005

[8] "A Survey of Anonymous Communication Systems", Mathew Edman, ACM, Computing Surveys, Vol. 42, No. 1, Article 5, December 2009.

[9] http://www.i2p2.de/how_garlicrouting, - accessed 25 March, 2011