Computers in Human Behavior 51 (2015) 780–791

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

A collaborative-based approach for avoiding traffic analysis and assuring data integrity in anonymous systems

Ramzi A. Haraty *, Bassam Zantout

Department of Computer Science and Mathematics, Lebanese American University, Beirut 1102 2801, Lebanon

ARTICLE INFO

Article history:
Available online 25 October 2014

Keywords:
Data integrity
Traffic analysis
Collaboration

ABSTRACT

The paper introduces a new collaborative technique for assuring data integrity while avoiding traffic analysis and other types of similar attacks (e.g., man-in-the-middle, data source fingerprinting, etc.). The new technique utilizes a quorum based approach to allow the client to validate the authenticity of the received data at his/her end by comparing different copies of the data. Similar to a reputation system, the new approach relies on the feedback of end-user communication experiences as well as a centralized entity to determine the trustfulness of nodes in the system. The new approach also is a hybrid of centralized and decentralized system that will help in keeping the system 'alive' to prevent different types of attacks that are carried out on centralized and decentralized peer-to-peer systems. The technique also accomplishes data transfer from source to destination using a distributed system, encryption, and a relatively new way for communication amongst system components.

1. Introduction

Since the day the Internet became a common and reliable mechanism for communication and data transfer, security officers and security enthusiasts rallied to enforce security standards on data transported over the globe. The goal was to achieve data integrity and confidentiality while using a reliable data transport medium. Whenever a user tries communicating with another recipient on the Internet, vital information is sent over different networks until the information is dropped, intercepted, or normally reaches the recipient. This information identifies where the request is coming from by revealing the user's IP; and hence, the geographical location, what the user needs from the recipient, and sometimes the identity of the user. The moment the recipient replies back, the same type of information is sent back along with a certain payload (meaningful content) for which the user had requested. Critical information traversing networks is usually encrypted. Sometimes encrypting the payload alone is not enough for users who wish to conceal their identities while communicating with recipients over the Internet. Take, for example, a reporter working undercover and sending critical information over the Internet to a country that is at war where the reporter is residing in. If the reporter's identity is revealed then the reporter's safety might be jeopardized. Hence, concealing who is sending the information is sometimes much more important than revealing the information itself. In order to conceal the sender's identity, different implementations have proven successful – one of which is the invention of anonymous networks (Scott, 2005). Anonymous networks go beyond transferring information over the Internet, whereby theoretically, the implementations can be replicated on different communication technologies such as mobile devices and wireless networks.

This paper presents a new technique that is inspired by many existing technologies used nowadays on the Internet. The new technique will not only use conventional methods for assuring data integrity but will also add a new approach for integrity validation that will be done on the client's end.

The remainder of this paper is organized as follows: Section 2 provides background information on anonymous systems. Section 3 introduces the new model. Section 4 presents and experimental results and Section 5 concludes the paper.

2. Background

Anonymous networks first emerged in the mid-1980s with a simple implementation of Chaum Mixes (Chaum, 1981) and the Anonymizer. Users connect to a single entity acting as a proxy that relays connections to different destinations. The identity of the sender is concealed but the destination is not; however, since hundreds of requests could be established from a single entity then pinpointing the source proved to be difficult. As adversaries gained interest in anonymous systems, many different scenarios, theories, and implementations have emerged for protecting the transmitted

* Corresponding author.
E-mail address: rharaty@lau.edu.lb (R.A. Haraty).

http://dx.doi.org/10.1016/j.chb.2014.09.031
0747-5632/© 2014 Elsevier Ltd. All rights reserved.

and received data (Danezis, Dingeldine, & Mathewson, 2003; Freedman, Sit, Cates, & Morris, 2002; JAP Anonymity and Privacy). Consequently, many different attack or counter-attack techniques have also emerged to challenge these security defenses. Smart deciphering, cracking of encrypted data, man-in-the-middle attacks, data replays, data-source fingerprinting, time attacks, and many others are all examples of what anonymous systems are subjected to currently (Ibrahim, Abuhaiba, & Hubboub, 2012; Ornaghi & Valleri, 2003; Whalen, 2001). Government organizations have also paid a great deal of attention to anonymous systems whereby the most commonly used anonymous system, Tor (based on second generation onion routing), is sponsored by DARPA and under the High Confidence Network Program as well the United States Navy (ONR) Haraty & Zantout, 2014. Additionally, some governments have reacted negatively to anonymous systems whereby these systems have now been banned from being used inside countries such as China, Saudi Arabia, and Germany for different reasons (China bans anonymous internet messages). As anonymous systems evolve, so is the understanding of the concept of anonymity by different computer user groups and societies in general.

The topic of anonymity has been the passion of many information security enthusiasts. However, the number was very little compared to researchers involved in other computer science topics at the time. Although the number of successful anonymous designs and implementations span to approximately 10 systems for which only two or three have been widely adopted, every system has its design flaws and features (Fernández Franco, 2012; Jansen, 2012; Ries, Panchenko, State, & Engel, 2011).

Throughout the past couple of years, scientists, researches and freedom activists have all been exposed to the topic of anonymous communication that can provide a sense of security to their identity on the Internet or during P2P communication. While public awareness has not been fully reached, research continues to take place and the topic of anonymity has been introduced as part of the curriculum to some of the leading universities in the west, as well as in Europe. To the anonymous community's surprise, some anonymous systems (like Tor, NetCamo, etc.) are being sponsored by government agencies, such as DARPA and the US Navy. This raises a lot of eyebrows and many questions to such security interests by governments in anonymous communication.

Developers of Tor, I2P, NetCamo and almost every anonymous system have clearly stated that their system cannot prevent against global adversaries, one of which are governments. Hence, should the whole concept of anonymity be cancelled and forgotten about? The question that should be asked is, how much are global adversaries interested as well as worried of concepts like anonymous communication and what is being done to strengthen/weaken or even alter this new awareness and scientific interest? More work is being put into coming up with the most advanced anonymous system that can prevent even against global adversaries and especially governments. This has lead governments such as Germany, China, Kingdom of Saudi Arabia, and others to ban the use of anonymous systems for the following reasons:

1. Governments need to monitor and control the use of the Internet communication for the sake of national security, and intelligence gathering.
2. Governments need to protect their people from being the victims of anonymous communication misuse.
3. Governments need to prevent against using anonymous systems as tools for terrorists and organized crime's undetectable communication.

The Chinese government, for example, chooses to ban Tor and other anonymous communication mainly because of national security. They simply do not wish to have information leak in or out of their country without the knowledge of Chinese intelligence agencies. There have been rumors where the Chinese government had cloned the Tor network at Internet gateways and while Chinese users think they are connecting to Tor, they are actually connecting to Chinese Tor proxies and then being routed to the outside Internet world. In addition, a Tor network is also being run inside China in order to camouflage communication between users inside the country. However, one can only wonder how secure this communication is! And whether or not it has been sponsored and introduced by the Chinese intelligence already.

The German government also banned the use of Tor as of January 1, 2008 because of the incidents and consequences inflicted by users of Tor. Since the source of anonymous communication cannot be tracked, then any message sent by a source can be completely concealed and the destination is unaware of who sent the original message, nor by whom it was relayed from, except for the last entity that delivered the message. Unfortunately, this may indicate to victims that it was the entity that delivered the message – the actual originator, which of course is not the case. As such, there have been two incidents where a bomb threat and kidnap ransom note were relayed through Tor exit nodes located in Germany which had been setup by innocent Tor enthusiasts. This had led the German authorities to accuse Tor enthusiasts of participating in such criminal acts. Accordingly, and after thorough investigations, the Tor anonymous network was to be fully banned from being used in Germany to protect the community from similar incidents.

It is clearly evident that anonymous networks have become terrorist and criminal magnets that attract malicious groups in relaying information from source to destination. It has also become used by embassies and some government agencies that choose to relay their traffic through anonymous networks in order not to be detected by any snooping party. Thus, one can deduce that an anonymous system is a two edged sword where it can be used for different conflicting purposes. Roger Dingledine, one of the core developers of Tor, had strongly argued with anonymous critics that criminals and terrorists have their own different means of communicating their plans; and hence, Tor does not present a mean for criminal use. Critics argue that as anonymous systems become sophistically complicated then Tor or other anonymous systems may become a reliable tool for criminals.

As social and government awareness arises, and as anonymous systems improve to protect against global adversaries for whom governments are part of, would anonymous systems survive? The key in anonymous system survival is to protect anonymous users from being the victims of incidents such as the ones aforementioned, and to prohibit illegal use of the system. Hence, anonymity has to be redefined to categorize different types of global adversaries for which governments, terrorists, freedom activists and/or other entities may or may not be part of. Incidents like the ones previously mentioned need to be dealt individually on a case by case basis. Then again who should define/control this categorization and release of critical information, as a terrorist in one country might be considered a hero in another; and this brings even more complication to this process where politics and national/international security become involved.

In order to encourage the use of a new anonymous system; and therefore, have it adopted and supported by all entities, one has to revisit the concept of security and freedom of communication in anonymous systems and realize that responsibilities do exist and that when a misuse occurs then malicious users need to be identified and either banned from further using the system or reported to authorities.

The methodology used in this work was inspired by four different implementations – BitTorrent (Zantout & Haraty, 2010), Tor, I2P (Zantout & Haraty, 2011), and NetCamo (Guan et al., 2001) for which the first three are widely used today and have made a major impact on the world of networking and particularly peer-to-peer communication. While BitTorrent is not considered an anonymous system, it is a unique mechanism for peer-to-peer communication. Tor's design and continuous active development, as well as the amount of users it has gathered over the years have made Tor the choice for any user wishing to conceal his/her identity over the Internet. Inspired by Tor, I2P utilizes the same concepts of Tor; however, it adds a further layer of security by separating the transmission and receiving streams in the anonymous system. NetCamo aims at delivering content for QoS demanding applications through different hops and networks (Choi, Xuan, Li, Bettati, & Zhao, 2000).

3. The model

This paper introduces a new technique to avoid traffic analysis while assuring the integrity of the data being transmitted from sender to recipient. Many researchers and security enthusiasts have rushed during the past several years to publish a number of documents supporting anonymous networks while detailing weak points and new techniques to remedy problems in existing anonymous systems (Bauer, McCoy, Grunwald, Douglas, & Tadayoshi, 2007; Chakravarty, 2014; Hints, 2002). Most of the publications have been focused around improving Tor (Second Generation Onion Router) or adding extra implementations or utilities around Tor in order to increase its performance and have a wider application compatibility with the Tor system (Murdoch, 2006). The reason behind the attention to Tor is because of three reasons:

1. Tor is supported by the United States Department of Defense and the US Naval Research office, with overwhelming amount of resources.
2. Tor has the simplest and easiest client setup for end users to install and use immediately on multiple platforms.
3. Tor was the first anonymous design that has been proven to work on the Internet and that has been around since the mid-1990s.

Hence, while it is easy to point out weak points in Tor, or any other anonymous system, and then execute a plan then an implementation for adding extra features that would enhance these systems, in our opinion the current designs and implementations still lack many important points. As a result, the work at hand aims at introducing yet another design for an anonymous system that has been inspired from the technologies and designs of different existing systems. Knowing that there is a tradeoff in any design or implementation, the new system design includes all the strengths of the existing anonymous system designs while introducing the concepts of 'Collaboration' and 'Data Integrity.

Up until now, a number of methodologies have been discussed that aim at preventing mostly traffic analysis with little guarantees of high availability as well as assuring the integrity of data being transmitted to recipients or even being transmitted by senders (Syvneron, Tsudik, Reed, & Landwehr, 2009).

The objective of this work is to avoid traffic analysis while assuring data integrity by using a collaborative based approach. Similar to a reputation system, the new approach relies on the feedback of end-user communication experiences as well as a centralized entity to determine the trustfulness of nodes in the system. The new approach also is a hybrid of centralized and decentralized system that will help in keeping the system 'alive' to prevent different types of attacks that are carried out on centralized and decentralized peer-to-peer systems. Additionally, a new communication scheme is presented in order to diversify and increase the randomness as well as the path of traffic generated by senders and recipients.

Since the inception of anonymous systems, the Internet was chosen to be the best test bed for experimenting with sending information to different entities anonymously. At first anonymous systems where meant to remain as closed circuits, for which senders and receivers are unaware of one another's identities. However, as some anonymous systems became adopted by end users, anonymous system circuits where no longer closed but now open for anonymous communication publicly over the Internet with other recipients who are unaware of such systems. Consequently, in order to protect and enhance the adoption of any existing and new anonymous system, a centralized tracking system is , that does not aim at revealing the content of communication streams, nor the inner communication between senders and recipients in closed circuits, but the entities involved in communication over the Internet through specific, trusted, and predefined exit nodes.

Although some critics might argue that this strictly contradicts with the definition of an anonymous system, however the argument here is that as long as communication occurs inside an anonymous system, in closed circuits, then anonymous users are protected and no logging of user information occurs. The moment anonymous communication leaves the closed circuit and now communicates publicly through trusted *exit* nodes is the moment logging of source and destination information starts being stored. This approach will hold any malicious user liable; and hence, can be tracked down in case of misconduct.

The logging system is viewed as a centralized system; however, its inner components are distributed amongst different storage hardware that is secured using encryption and restricted communication; and hence, the complete set of information about user identities cannot be revealed if a single machine is breached or compromised. The distributed storage system can be a distributed DBMS that is capable of storing different parts of information about the senders' details on different nodes running the DBMS. The storage hardware need not be geographically distributed; however, it is not advisable as DBMS intercommunication overhead might occur. For further growth requirements, a geographically distributed cluster of DBMS clusters can be created for performance and redundancy at later stages when logging load becomes high. There are many designs and technologies for distributed and secure DBMS implementations (Coy, 2008). However, these are outside the scope of this paper.

The aim of the logging system is to reveal preliminary information about the identity of a user in order to provide authorities with the bare minimum to identify malicious users. The logging system will also be capable of controlling *exit* nodes and store an Access Control List (ACL) of who is and is not allowed external communication. What is meant by the term *trusted* is *controlled* and *secured* nodes that can be trusted by anonymous users, whereby control and security procedures are executed by system administrators of the system.

Unlike Tor or other anonymous systems, access to the Internet through public nodes is disallowed. The decision/design has been taken due to incidents where malicious users would run large numbers of dedicated exit node servers; and therefore, snoop on the outgoing communication being transmitted in the clear to public servers on the Internet. Although the communication stream does not reveal the identity of the sender; however, the identity can be deduced from the content of the data being transmitted. A Swedish security officer for example was able to reveal the usernames and passwords as well as the messages of almost 100 embassies using Tor, while running only four dedicated Tor exit nodes (McCoy, Bauer, Grunwald, Kohno, & Sicker, 2008). For this reason, the proposed model chooses to route all communication with the Internet through trusted exit nodes that aim at protecting the user information from being snooped on, as well as protecting against malicious use.

The steps below detail the algorithm of how an anonymous node can request communication with other public servers on the Internet through trusted exit nodes:

1. Whether a node is new or has previously logged on to the anonymous system, the node can immediately communicate with other nodes on the system without contacting the centralized logging system.
2. The moment a guest node requires communication with an Internet server node, the guest node sends an encrypted message request to one of the centralized nodes, using the latter's published public key, and coupled with a newly generated public/private key by the guest node. The message is camouflaged, padded, and encrypted in order to conceal the request for public communication. Additionally, the guest node never sends the request directly to the centralized node, but actually has to relay it through a random node of the guest node's choice which is the bridge node only. This relaying through the bridge node is done for two reasons:
   a. In order to camouflage and hide the fact that the guest node needs to communicate with the centralized system. Hence, attackers do not realize that communication exiting the anonymous system is about to occur.
   b. In order to log the IP addresses of both the bridge node and the guest node as the communication moves along anonymous channels - the information about the identity of the sending receiving party gets stripped away from the packets/cell being sent and received. Hence, at any time, a node in the system is only aware of the next node it needs to forward to, and the previous node it received the data from. The reason behind logging both bridge and guest node IPs is to ensure that the two nodes are not under the control of the same entity; and hence, may spoof the IP of the guest node. In any case, the centralized system will reply back through the bridge node and both entities will be recorded in the logging system.
3. For identity concealment reasons, it is always preferred that the guest node generates a new set of public/private encryption keys once it needs to communicate with the centralized nodes. This is because the public key, provided by a recipient, is logged in the centralized logging servers; and hence, multiple entries may reveal different requests and times for accessing public servers. Had the public key for the guest node been static then this can be coupled with other attacks to reveal the identity/history of a certain node since the public key becomes its identifier.
4. The centralized node decrypts the camouflaged request and then logs the guest node's IP address as well as the bridge's address and then replies back to the guest node through the bridge node, with a message that the request has been processed as well as a 2048 bit string identifier and a timestamp for which external access will expire and another request needs to be sent to the centralized system again. The centralized logging system then sends encrypted notifications to all trusted exit nodes in order to accept communication and log requests by the guest node to the centralized system. A copy of the information, sent to the recipient, is also distributed to all exit nodes that are able to translate this information into an access control list. Of course all communication occurs through encrypted cells using public and private keys. The 2048 bit string will be required to be bundled with every request sent by the recipient to the trusted exit nodes. Since the exit nodes in the system are unable to verify the IP of the sender then an exit node will verify the identity using timestamp originally sent by the logging system as well as the public key of the recipient which are all stored as part of the ACL in every exit node. This design adds a layer of security for a number of attacks. Precisely, three forms of indirect attacks can take place on the centralized logging system, two by the anonymous user wishing to communicate with public servers, and the other by eavesdroppers waiting to detect exactly when a request for communicating with the outside world is requested by a certain users. The following is a brief explanation of the attacks:
   a. A malicious user, using one guest node, contacts the centralized logging system through a bridge node, and then forwards all the information received by the centralized system to second guest node in the anonymous system. The second guest node now holds all the information of the first guest node and the system recognizes the traffic exiting the system as the traffic being sent and received by the first guest node. When preventing against skillful attackers, adding complex strategies for insuring that the system is not cheated is necessary. Accordingly, the way to prevent against this type of attack is to allow the centralized logging system to perform a network latency test, again through bridge node, in order to perform a challenge request. A challenge request is nothing but a heartbeat with a certain question that needs to be replied to as soon as it has been received. In this scenario the centralized logging system as well as the exit nodes will coordinate a certain challenge whereby the answer is sent by the exit node to the guest node, while being padded randomly in one of the cells, and the challenge request is sent by the centralized system directly to the bridge node and then to the first node. If the first guest node is indeed malicious and even if an automated system has been developed by the attacker to forward messages to the second malicious guest node, then a certain delay will occur. This is due to the fact that the first guest node needs to forward the challenge to the second guest node and then the second node will need to reply back to the first node with the answer sent to it by the exit node, and then the first node should relay this information back to the centralized system. The tremendous delay in this action will determine if there is foul play. The challenge and response technique need not be predefined but randomly injected during guest node communication with the exit nodes.
   b. Although this is unfortunately still possible nowadays due to the carelessness of some security and networks administrators, IP spoofing is a problem that is handled well by our proposed design. The problem with IP spoofing is that incorrect Internet address information is sent to the centralized system, which automatically gets recorded and confirmed. However, the prevention for such an attack does not lie only in the centralized system alone but mainly in the bridge node itself. Given that a certain attacker fakes the IP of his guest node and then tries communicating with a bridge on the network, then the spoofed IP is detected via the anonymous system handshake technique that requires the IP to be valid. The bridge cannot but validate the IP of the guest node by establishing an anonymous circuit with the guest node. If the guest node replies back, then a legitimate connection is established; otherwise, the request is discarded by the bridge node. Consider the case where both the bridge node and the guest node are under the control of an attacker. Then the bridge node will actually accept the guest node's spoofed IP address and should forward it to the centralized system which in turn will never detect that anything wrong has occurred. The request for accessing the public network will be granted through exit nodes and even challenge requests cannot detect this attack. However, since the centralized system logs both information

about the bridge node and the guest node in any public communication, then both of these nodes are held liable in case of misconduct. Hence, the system can and still prevents against forged identities of nodes.

c. If a snooping party manages to detect that a guest node has sent a request for communicating publicly through exit nodes, and then the snooping party can also manage to inject a number of nodes which may act as bridge nodes that in turn can redirect or capture traffic sent from and to the central logging system in addition to the guest node through of the bridge node, then this causes no harm to the guest node because:
    i. Public keys of the centralized logging system are known; hence, no identity forgery can ever take place.
    ii. All traffic is encrypted with public and private keys of the guest node and the centralized logging system.
    iii. During communication multiple paths are chosen using different sessions as they are spawned by the end user; hence, the chance of snooping on the whole connection becomes difficult.
    iv. Padding and camouflaging of traffic always takes place.
    v. Traffic to the public network always goes out of the system through random exit nodes and not a single exit node that can be monitored by the snooper; and hence, timing attacks cannot take place.

As one can observe, the tight design of the system is simple as well as efficient in detecting anomalies.

4. Experimentation and testing

It is interesting to note that similar testing for anonymous systems has been carried out in real live environments and not fully simulated. Tor, for example, had started with three Sun servers with multiple instances running on the same machines. Currently, there are entities that are providing large scale test beds with 500+ nodes that are distributed along the five continents where anyone can reserve and rent the servers on a daily basis. The cost, however, is overwhelming. The concept of testing our system is to simulate that there are multiple *exit* nodes deployed all over the globe, and that these nodes are actually relaying data from/to nodes sitting inside the anonymous system. Given that the nodes relay traffic and send summaries to the centralized logging system, the goal is to record whether or not it is possible to accomplish such information logging, given the huge bandwidth *exit* nodes handle.

Fig. 1 below shows the five continents as well as exit nodes marked in red dots, and an active centralized logging system in the middle of Europe. The straight yellow lines simulate a conceptual communication with the centralized logging system although communication can be sent indirectly through other anonymous relay nodes in the system.

The rate of logging information being passed to the centralized logging system is extremely important because if logging causes an overhead on the system, then this will affect all communication exiting and entering the anonymous system. Of course it is impossible to attain such a huge number of equipment to generate so much traffic, yet again the traffic being generated can be simulated and the logging system can be built and tested with enough hardware and a sponsor. Luckily the implementation and testing of the system has been completed using a product from Cisco utilized for a slightly similar purpose. The *Service Control Engine* (SCE) from Cisco is a network appliance that is utilized to prioritize and/or monitor traffic in carrier grade telecommunication corporations as well as huge Internet Service Providers. The implementation and testing was done at the National Engineering Services and Marketing Company Ltd. (NESMA) in the Kingdom of Saudi Arabia. NESMA is a group of companies with over one billion US dollars of yearly revenue and one of the companies in NESMA's group is NESMA Internet which happens to be the second largest Internet service provider in KSA. All experimentation and testing was carried out during a period of two weeks at NESMA Internet whereby dedicated resources were provided for the testing phase. The SCE appliance from Cisco is a carrier-grade high capacity appliance, capable of being installed inline in any network (Introduction to the Cisco). The appliance is able to run in promiscuous mode in order to inspect or alter the traffic behavior. The SCE is then able to convert all Internet transaction information being logged, into SQL queries capable of being fed to high performance SQL servers whereby reports can be generated for utilization and marketing purposes. The aim of this experimentation is to acknowledge whether or not a certain amount of user's traffic can be logged in real time to a centralized logging system. The system can be decentralized internally using distributed SQL server technologies. The aim is also to measure how much bandwidth is generated from each *exit node* in the model and how much disk storage is required especially that archiving needs to take place. This information is critical to know, in order to learn whether or not the anonymous logging system can be successfully deployed on a larger scale. For the experimentation and testing of this section the following was needed and was met by NESMA's resources:

1. A portion of random speeds of Internet DSL users.
2. The ability to convert the traffic generated by users to Internet transactions, capable of being logged to (a) system(s) to SQL queries that represent traffic summaries.
3. The data being generated is easily retrieved and distributed in order to ensure that if a single machine is compromised then the logged data is not revealed to the attacker.

For the first point, exactly 2956 DSL subscribers of random Internet speeds ranging from 256 Kbits/s to 1 Mbit/s where selected generating traffic of approximately 138 Mbits/s on a daily average. The exact amount of speed ratios for users is unknown due to random and frequent subscriber joins and departures on the system. The users have been selected also at random to be normal home users and non-governmental or business/professional companies. In other words, the amount of traffic generated is unrestricted end-user traffic that includes all sorts of protocols such as P2P traffic as well as VoIP and HTTP. One of the characteristics of the SCE appliance is its ability to convert traffic sniffed of the network to meaningful and summarized data (transactions) sent to separate SQL servers on the fly via SQL queries. It is required that the SCE appliance and the SQL servers be of powerful hardware in order to sustain and withstand the amount of load they are subjected to. The network design and implementation for the above solution has been integrated with NESMA's data center infrastructure. All network devices and appliances are connected to each other via Fiber Channel and/or 1Gbit Ethernet using CAT6 connections. Fig. 2 illustrates how DSL users connect to the Cisco 7200 router (authentication process has been omitted for the sake of simplicity), and then all the traffic generated from and to the DSL users passes through the Cisco SCE 2000 box that is controlled in this experiment by two servers. The SCE appliance also passes the traffic to users through Fiber Channels to the Cisco 6500 switch and then to the core router that is in turn connected to one of the Internet Backbones. Traffic is intercepted, and every network packet is accounted for by the SCE appliance for outgoing and incoming connections. The appliance will then communicate with

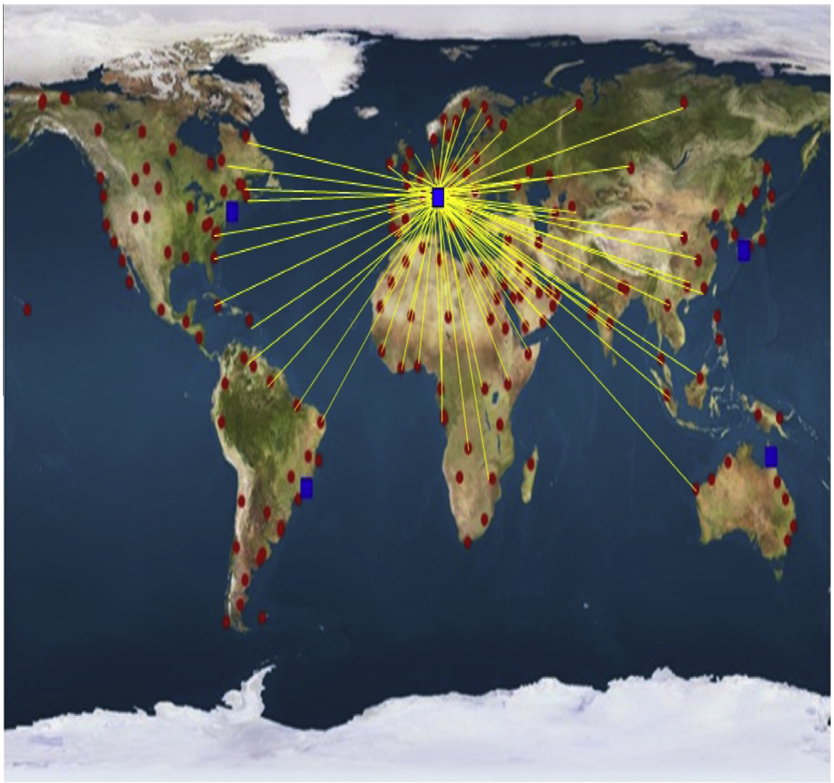


Fig. 1. Different geographically distributed nodes in all continents.

the management servers to issue SQL queries communicating over the Internet.

As previously mentioned, the logging system in the theoretical model needs to be centralized from the exit nodes' perspective, but the data is internally distributed amongst different number of nodes for security reasons. Hence, a distributed storage engine, that can store distributed data amongst different nodes, is needed and therefore the first solution that comes to mind is a distributed DBMS cluster. Whenever data needs to be retrieved then the database administrator can query a number of nodes for retrieving the secure information. Additionally, in order to relate this network design to the world map shown previously, one can imagine that the SCE appliance as well as the 7200 Cisco router as a single entity that represents an *exit* node or maybe many exit nodes since the amount of bandwidth allocated for the users is considerably high. The DSL users can be considered as users in the anonymous system relaying traffic through the exit node (Cisco appliances) and then trying to communicate with other hosts on the Internet. The Database servers are the centralized logging system that logs all information securely. Fig. 3 illustrates a simpler version of the preceding network diagram relating to what has just been explained.

Since exit nodes are expected to communicate with a single entity in secure communication, then the DBMS distributed cluster needs to be represented with only one service that can accept SQL data. This can be accomplished using a relatively new technique in MySQL DBMS called an SQL proxy (Create a SQL Server Agent Proxy). The SQL proxy acts as a centralized entity that accepts any incoming SQL transactions. Behind the SQL proxy is a number of SQL nodes that are configured for performance and reliability. Hence a farm of SQL nodes or a cluster of nodes can be configured behind the SQL proxy whereby the proxy relays information from and to these nodes. The concept is to create an abstraction layer on the physical level so that queries are directed only to the SQL proxy without worrying about how the SQL query is executed and who executes the SQL query.

It is the SQL proxy's job to ensure the availability of the SQL nodes sitting behind it and hence sending to the correct and available/resource-free nodes for faster performance. If a problem occurs then the SQL proxy is also capable of queuing incoming SQL queries until the backend system is up again. Of course the SQL proxy can also be set up in high availability mode so that no single point of failure exists. For simplicity reasons, two powerful quad dual-core processor servers have been chosen to be deployed in high availability active–passive configuration that represents the centralized logging system. The servers are connected via Ethernet to the Cisco 6500 switch that communicates with the Cisco SCE equipment. An SQL proxy has not been used in this scenario mainly because the number of active servers is one at any point in time (due to the active–passive HA technique used). It is important to note that the SQL proxy can exist on a third machine and redirect all SQL traffic from and to the SCE appliance. Accordingly, in a real life example where a centralized system is required and in order not to expose the centralized system to the public and have the risk of exposing the set of IPs which might get the system attacked or even blocked by global adversaries, a proxy or a similar functionality can be introduced for which can be rotated amongst different trusted nodes. This trusted proxy node, or multiple trusted proxy nodes, can at one time communicate with the centralized system without exposing to end-users where the system is actually located.

The SQL schema used for storing information about traffic usage of anonymous users wishing to exit the anonymous system has been selected from a similar existing schema for the SCE appliance original designed by Cisco. After interconnecting all components and making sure that everything would work correctly when

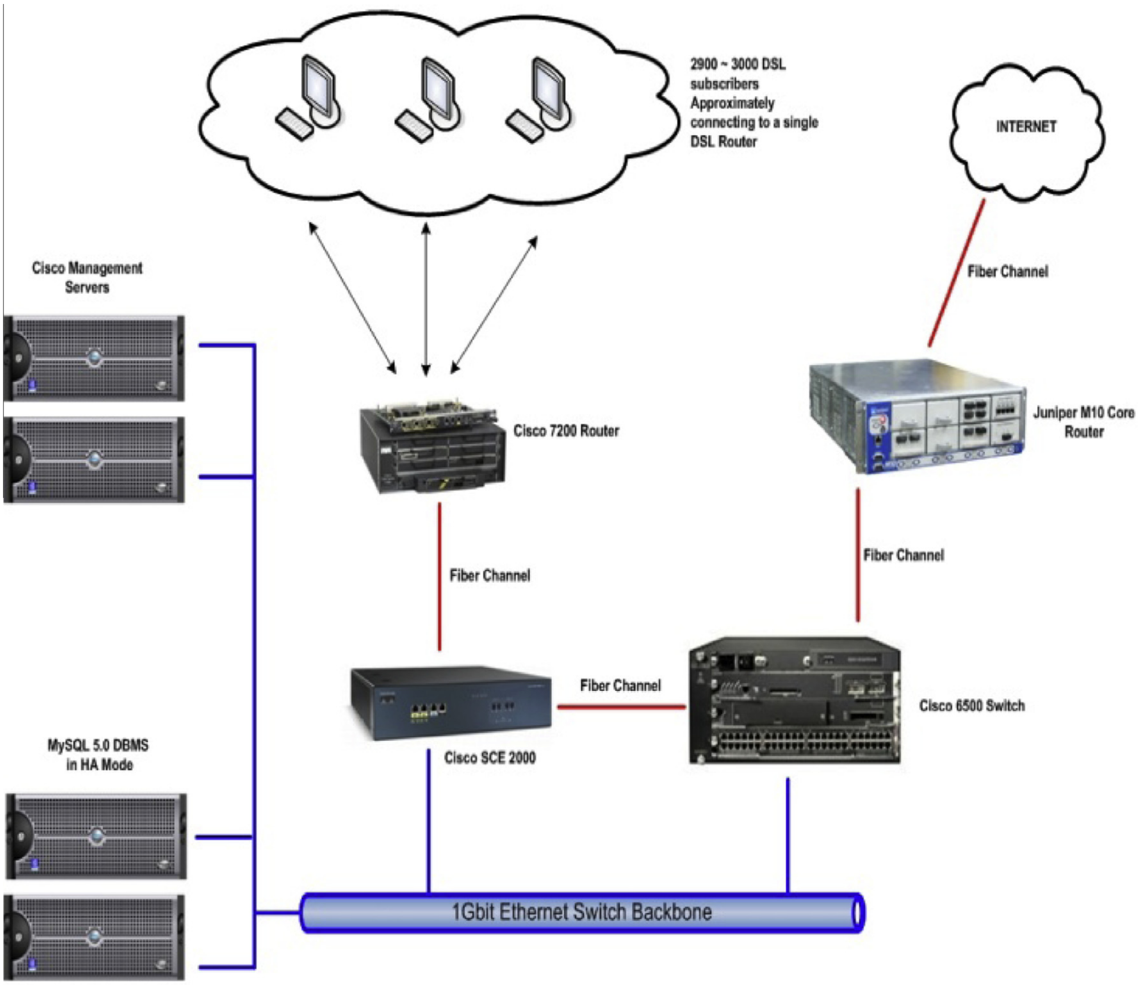


Fig. 2. The experimental setup.

the green light is given for logging traffic, the systems was turned on.

4.1. Experimental results

Testing has been carried out in order to:

1. Learn whether or not huge amounts of data can be logged to the system in summarized information (transactions) such as SQL queries.
2. Measure how much bandwidth is generated by an exit node while feeding the logged information to a centralized system.
3. Measure the amount of data generated by the system in order to cater for archiving and disk storage.

The system was left logging information for a period of seven days and measurements at peak time and non-peak time have been taken at different intervals daily. The following points answer the above questions:

1. It is definitely possible to log the information being generated by 2956 connected DSL users on average with numerous amounts of protocols being used. DSL users were consuming a combined speed of inbound and outbound connection of 132.36 Mbits/s on average with approximately 159.43 Mbits/s during peak times. Section two and three below elaborate more on this. Fig. 4 presents the total consumption of inbound and outbound traffic utilization on the single Cisco 7200 router. The graph was taken using SNMP polling on the router and "rrd-tool version 1.2" software for graphing. The router was polled every 5 min for a duration of a week in order to come up with the values below.

In order to better understand this, one can imagine that the SCE appliance and the Cisco 7200 router as a single entity acting as traffic relay as well as traffic logger for data being sent and received by end-users. All the logged information is sent to a centralized system, which is the DBMS.

2. The amount of summarized information (Internet transactions executed by users) being sent over the network as SQL queries, generated by the SCE equipment and sent to the DBMS engines, is approximately 192 Kbits/s on average. This is the summary relative to the 132.36 Mbits/s passing through the system at any point in time.
3. The number of SQL queries pertaining to the 192 Kbits/s generated by the SCE equipment is 110 SQL insert queries per second. This enlarges the above schema size by 2.08 Gbytes/day and hence the total storage required for this experiment was approximately 14.2 GBytes for seven days and a total of 66,529,538 queries where executed on the DBMS.

It is important to note that the SQL servers as well as the SCE appliance were utilizing 5% of total CPU usage on average as the

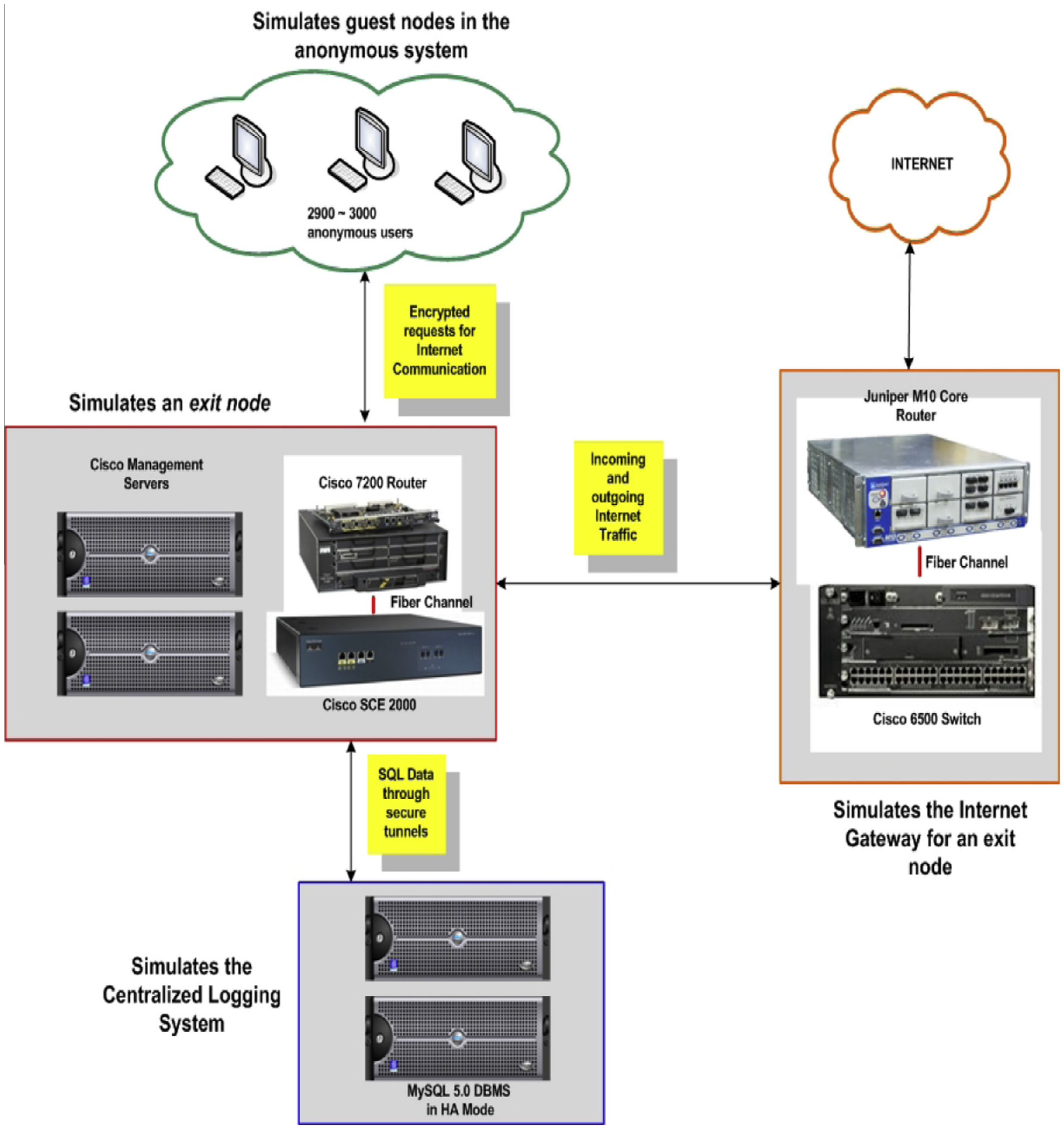

Simulates guest nodes in the anonymous system
2900 ~ 3000 anonymous users
INTERNET
Encrypted requests for Internet Communication
Simulates an exit node
Cisco Management Servers
Cisco 7200 Router
Fiber Channel
Cisco SCE 2000
Incoming and outgoing Internet Traffic
Juniper M10 Core Router
Fiber Channel
Cisco 6500 Switch
Simulates the Internet Gateway for an exit node
SQL Data through secure tunnels
Simulates the Centralized Logging System
MySQL 5.0 DBMS in HA Mode

Fig. 3. Different logical representations of equipment.

servers and the appliance are ready to handle 2 Gbit/s of throughput. Hence, resource bottlenecks do not exist in this case. Moreover, the majority of the queries, as expected, being sent to the SQL servers were SQL insert queries. Fig. 5 presents a sample output of what was recorded for only HTTP usage in 2 min. The logs have been retrieved in the form of an SQL query directly executed against the database. Exactly 1177 entries were recorded.

One may notice that times are in decimal format, as they represent the UNIX timestamp of the aforementioned dates, this goes as well for IPs for easier search. The query provided this result had already been supplied with converted values. To better understand the above result, suppose that we have been informed that there was a misuse for a website coming from an *exit node* X for the domain i.ytimg.com on HTTP port 80. This information can be queried immediately on the *exit node* X resulting in the following:

Peer IP: 1208933808 in decimal or 72.14.221.176 in IP format.
Peer Port: 80
URL Appended to Domain: /vi/6CQ19ZDY2J4/default.jpg
Source IP: 89.5.220.163
Source Port: 56271
Millisecond Duration: 10,300 ms
Upstream Volume: 774

4.2. Remarks and observations

In our new model, it is expected that multiple exit nodes exist and each would be communicating with the centralized system. Hence, the amount of information being fed into the logging system might increase dramatically. However, it is important to recall that the users in the experiment, although might reflect the exact behavior of end-users on the internet engaging in P2P activities as well as utilizing other protocols, might use a much lower traffic rate since anonymous users might be much aware about the anonymous systems' traffic. Moreover, with traffic prioritization and traffic shaping techniques, the amount of traffic can be minimized; although this really is not a vital factor since hardware and band-

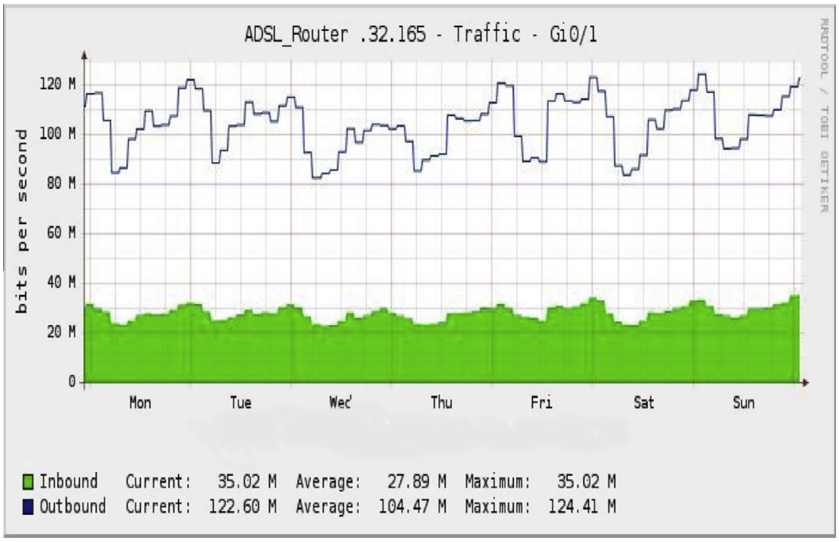


Fig. 4. Router graph for bandwidth egress/ingress utilization.

width has become a cheap commodity in the IT world. As a result, and in order to come up with a clearer picture with some tangible numbers, the Tor anonymous system has been taken as an example to compare the values presented by its system with the numbers of the new model. As documented on the Tor website, approximately 100,000 users are connected to the Tor network for which 960 Mbits/s are generated by all nodes on the system including exit nodes, relay nodes, and guest (end-user) nodes. According to these figures and the ones generated by the previous experiment, Table 1 is drawn to show the rate of data growth in the system and the requirements for Tor if it were to log all its traffic and not only at *exit nodes*.

In reality not all data is going to be logged since data exiting the anonymous system is going to be analyzed by exit gates and summary logs are going to be sent to the centralized logging system. Internal anonymous communication is never logged. Tor has 900 dedicated exit nodes in its system maintained by Tor systems administrators and enthusiasts. If those nodes were to be considered as dedicated trusted nodes in the new model, then it would be interesting to calculate how much throughput is required from each dedicated node assuming that the 960 Mbits/s bandwidth requirement becomes for Internet communication only, as shown in Table 2.

As one can notice that the amount of traffic per exit node according to the above table is actually low whereby dedicated exit nodes usually contribute 10 times more the amount of bandwidth on average. Even with 10 times the numbers above, these numbers are really negligible when considering server grade hardware whereby a quad processor server can withstand huge amounts of data being inserted in a DBMS. Additionally, the system is not being queried regularly as the majority of the SQL queries are actually summarized logs of external Internet traffic.

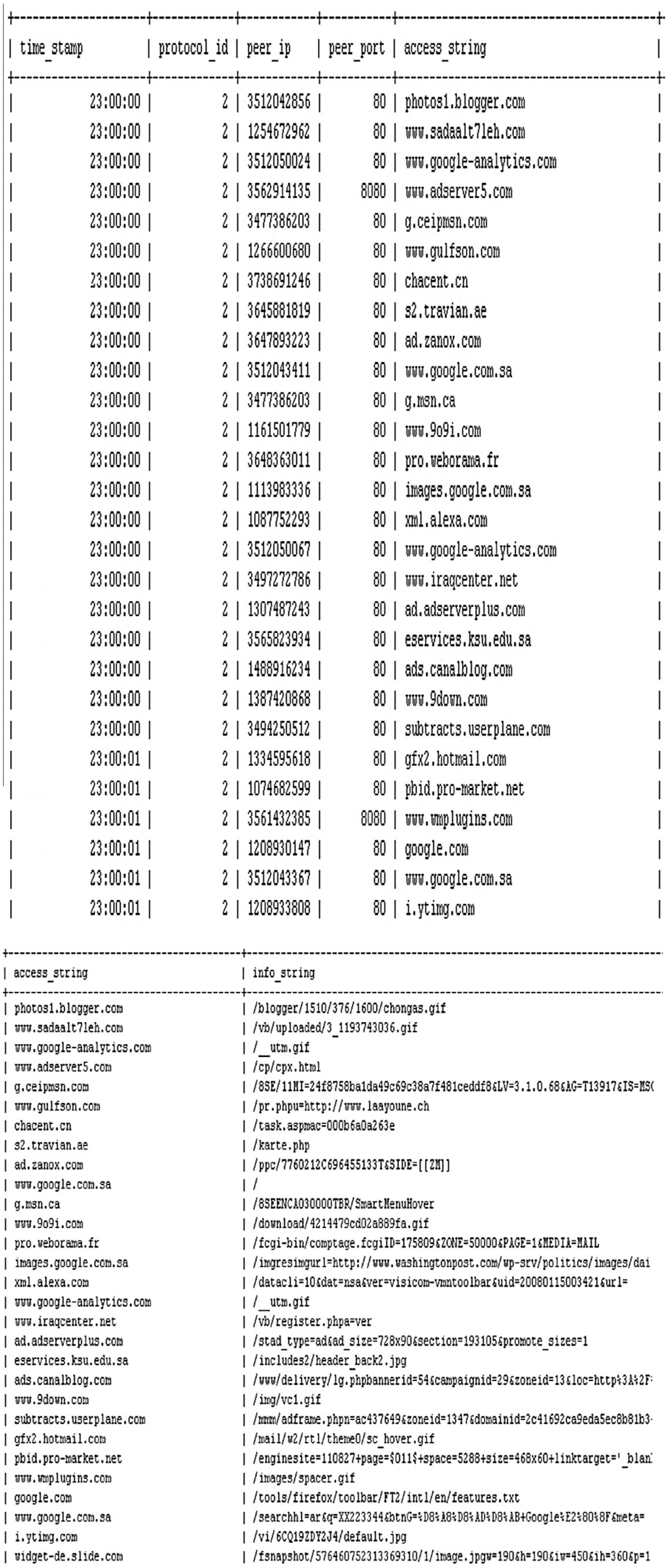
Assuring data integrity through checksums and encryption is an efficient and reliable method that ensures that the data being received or sent is indeed authentic. However, this is only acceptable when two users, in a closed network, are sending information to one another and a third party is conducting malicious behavior along the traffic stream. The integrity of data can still be compromised using different attack methods.

Regardless of the technique being used to compromise the data, since the data is being sent through a single stream then data integrity cannot be assured even though data is encrypted. Relying solely on encryption may prove fruitful in most cases; however, there are exceptions as shown in the previous example. I2P developers chose to implement their system similar to a packet switching model; however, attacks can be even conducted to comprise the content of data. Accordingly a new technique needs to be devised in order assure data integrity in traffic analysis avoidance systems. Our aim is to gather the feedback from different users in the anonymous system about the circuits they are connected to. If a user connects to a circuit where one node is misbehaving then the circuit is destroyed and the nodes forming this circuit are reported to the centralized system by the end-user. Although some non-malicious nodes may be mistakenly categorized as malicious as they are reported; however, the more the number of users report more suspicious nodes on a circuit, then only the nodes that will be continuously reported will gain high malicious points and the rest will be discarded or will be temporarily monitored for more reports.

We assume that all user nodes are actually trusted nodes that will truthfully report their experiences while connecting to different circuits. Moreover, the reputation system only takes into account reporting bad circuits in case of slowness experiences or denial of service; however, it does not cater for bad content or integrity checks on the delivered payloads. It is only late when a malicious node is detected as the data would have already been compromised or stored for analysis. Nevertheless, a reputation system is still necessary in avoiding communication with suspicious nodes when the data critical and should not be revealed.

We also assume that a number of exit nodes which are geographically distributed are trusted. However, none of the pathways from and to the exit nodes to end-users and to the Internet can be trusted. In such a scenario even if an exit node is trusted then all its traffic can be tampered with as it is known to global adversaries; hence, a chance for a compromise may exist. In order to limit this type of attack and add more certainty and security to the content being delivered to the anonymous user, a new implementation is needed. The new implementation aims at verifying that the data that has been collected from a server node on the Internet is indeed the data from the source as requested by the end user, and secondly that the data delivered to the end user can be verified by the end user, via a quorum approach, as being the correct data that has not been tampered with and is truly from the source.

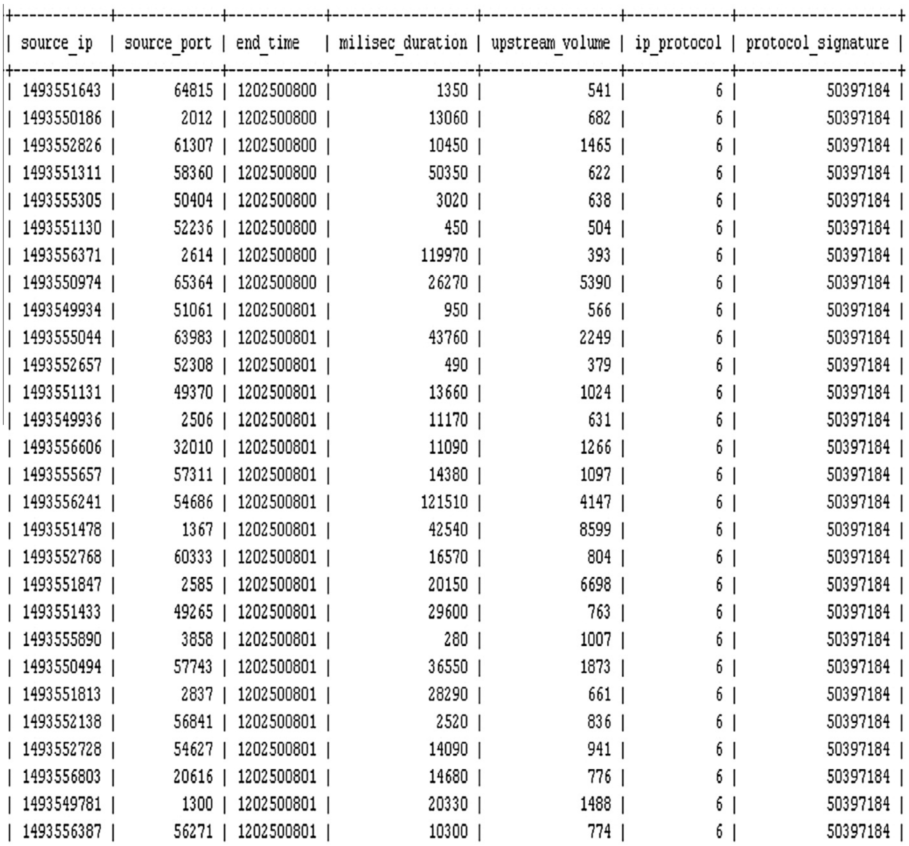
If a trusted node's communication stream is monitored or is being tampered with then the only way to verify that the content being downloaded from the Internet is authentic is by asking another trusted node to obtain the same content and then compare



```
+---------------------+-------------+------------+-----------+-----------------------------------------+
| time_stamp          | protocol_id | peer_ip    | peer_port | access_string                           |
+---------------------+-------------+------------+-----------+-----------------------------------------+
|            23:00:00 |           2 | 3512042856 |        80 | photos1.blogger.com                     |
|            23:00:00 |           2 | 1254672962 |        80 | www.sadaalt7leh.com                     |
|            23:00:00 |           2 | 3512050024 |        80 | www.google-analytics.com                |
|            23:00:00 |           2 | 3562914135 |      8080 | www.adserver5.com                       |
|            23:00:00 |           2 | 3477386203 |        80 | g.ceipmsn.com                           |
|            23:00:00 |           2 | 1266600680 |        80 | www.gulfson.com                         |
|            23:00:00 |           2 | 3738691246 |        80 | chacent.cn                              |
|            23:00:00 |           2 | 3645881819 |        80 | s2.travian.ae                           |
|            23:00:00 |           2 | 3647893223 |        80 | ad.zanox.com                            |
|            23:00:00 |           2 | 3512043411 |        80 | www.google.com.sa                       |
|            23:00:00 |           2 | 3477386203 |        80 | g.msn.ca                                |
|            23:00:00 |           2 | 1161501779 |        80 | www.9o9i.com                            |
|            23:00:00 |           2 | 3648363011 |        80 | pro.weborama.fr                         |
|            23:00:00 |           2 | 1113983336 |        80 | images.google.com.sa                    |
|            23:00:00 |           2 | 1087752293 |        80 | xml.alexa.com                           |
|            23:00:00 |           2 | 3512050067 |        80 | www.google-analytics.com                |
|            23:00:00 |           2 | 3497272786 |        80 | www.iraqcenter.net                      |
|            23:00:00 |           2 | 1307487243 |        80 | ad.adserverplus.com                     |
|            23:00:00 |           2 | 3565823934 |        80 | eservices.ksu.edu.sa                    |
|            23:00:00 |           2 | 1488916234 |        80 | ads.canalblog.com                       |
|            23:00:00 |           2 | 1387420868 |        80 | www.9down.com                           |
|            23:00:00 |           2 | 3494250512 |        80 | subtracts.userplane.com                 |
|            23:00:01 |           2 | 1334595618 |        80 | gfx2.hotmail.com                        |
|            23:00:01 |           2 | 1074682599 |        80 | pbid.pro-market.net                     |
|            23:00:01 |           2 | 3561432385 |      8080 | www.wmplugins.com                       |
|            23:00:01 |           2 | 1208930147 |        80 | google.com                              |
|            23:00:01 |           2 | 3512043367 |        80 | www.google.com.sa                       |
|            23:00:01 |           2 | 1208933808 |        80 | i.ytimg.com                             |
```

```
+-----------------------------------------+-----------------------------------------------------------------------
| access_string                           | info_string
+-----------------------------------------+-----------------------------------------------------------------------
| photos1.blogger.com                     | /blogger/1510/376/1600/chongas.gif
| www.sadaalt7leh.com                     | /vb/uploaded/3_1193743036.gif
| www.google-analytics.com                | /__utm.gif
| www.adserver5.com                       | /cp/cpx.html
| g.ceipmsn.com                           | /8SE/11MI=24f8758ba1da49c69c38a7f481ceddf8&LV=3.1.0.68&AG=T13917&IS=MS(
| www.gulfson.com                         | /pr.phpu=http://www.laayoune.ch
| chacent.cn                              | /task.aspmac=000b6a0a263e
| s2.travian.ae                           | /karte.php
| ad.zanox.com                            | /ppc/7760212C696455133T&SIDE=[[ZM]]
| www.google.com.sa                       | /
| g.msn.ca                                | /8SEENCA030000TBR/SmartMenuHover
| www.9o9i.com                            | /download/4214479cd02a889fa.gif
| pro.weborama.fr                         | /fcgi-bin/comptage.fcgiID=175809&ZONE=50000&PAGE=1&MEDIA=MAIL
| images.google.com.sa                    | /imgresimgurl=http://www.washingtonpost.com/wp-srv/politics/images/dai
| xml.alexa.com                           | /datacli=10&dat=nsa&ver=visicom-vmntoolbar&uid=20080115003421&url=
| www.google-analytics.com                | /__utm.gif
| www.iraqcenter.net                      | /vb/register.phpa=ver
| ad.adserverplus.com                     | /stad_type=ad&ad_size=728x90&section=193105&promote_sizes=1
| eservices.ksu.edu.sa                    | /includes2/header_back2.jpg
| ads.canalblog.com                       | /www/delivery/lg.phpbannerid=54&campaignid=29&zoneid=13&loc=http%3A%2F
| www.9down.com                           | /img/vc1.gif
| subtracts.userplane.com                 | /mmm/adframe.phpn=ac437649&zoneid=1347&domainid=2c41692ca9eda5ec8b81b3
| gfx2.hotmail.com                        | /mail/w2/rtl/theme0/sc_hover.gif
| pbid.pro-market.net                     | /enginesite=110827+page=$011$+space=5288+size=468x60+linktarget='_blan
| www.wmplugins.com                       | /images/spacer.gif
| google.com                              | /tools/firefox/toolbar/FT2/intl/en/features.txt
| www.google.com.sa                       | /searchhl=ar&q=XX223344&btnG=%D8%A8%D8%AD%D8%AB+Google%E2%80%8F&meta=
| i.ytimg.com                             | /vi/6CQ19ZDY2J4/default.jpg
| widget-de.slide.com                     | /fsnapshot/57646075231336931O/1/image.jpgw=190&h=190&iw=450&ih=360&p=1
```

Fig. 5. Sample report for HTTP usage.



```
+------------+-------------+------------+------------------+-----------------+-------------+--------------------+
| source_ip  | source_port | end_time   | milisec_duration | upstream_volume | ip_protocol | protocol_signature |
+------------+-------------+------------+------------------+-----------------+-------------+--------------------+
| 1493551643 |       64815 | 1202500800 |             1350 |             541 |           6 |           50397184 |
| 1493550186 |        2012 | 1202500800 |            13060 |             682 |           6 |           50397184 |
| 1493552826 |       61307 | 1202500800 |            10450 |            1465 |           6 |           50397184 |
| 1493551311 |       58360 | 1202500800 |            50350 |             622 |           6 |           50397184 |
| 1493555305 |       50404 | 1202500800 |             3020 |             638 |           6 |           50397184 |
| 1493551130 |       52236 | 1202500800 |              450 |             504 |           6 |           50397184 |
| 1493556371 |        2614 | 1202500800 |           119970 |             393 |           6 |           50397184 |
| 1493550974 |       65364 | 1202500800 |            26270 |            5390 |           6 |           50397184 |
| 1493549934 |       51061 | 1202500801 |              950 |             566 |           6 |           50397184 |
| 1493555044 |       63983 | 1202500801 |            43760 |            2249 |           6 |           50397184 |
| 1493552657 |       52308 | 1202500801 |              490 |             379 |           6 |           50397184 |
| 1493551131 |       49370 | 1202500801 |            13660 |            1024 |           6 |           50397184 |
| 1493549936 |        2506 | 1202500801 |            11170 |             631 |           6 |           50397184 |
| 1493556606 |       32010 | 1202500801 |            11090 |            1266 |           6 |           50397184 |
| 1493555657 |       57311 | 1202500801 |            14380 |            1097 |           6 |           50397184 |
| 1493556241 |       54686 | 1202500801 |           121510 |            4147 |           6 |           50397184 |
| 1493551478 |        1367 | 1202500801 |            42540 |            8599 |           6 |           50397184 |
| 1493552768 |       60333 | 1202500801 |            16570 |             804 |           6 |           50397184 |
| 1493551847 |        2585 | 1202500801 |            20150 |            6698 |           6 |           50397184 |
| 1493551433 |       49265 | 1202500801 |            29600 |             763 |           6 |           50397184 |
| 1493555890 |        3858 | 1202500801 |              280 |            1007 |           6 |           50397184 |
| 1493550494 |       57743 | 1202500801 |            36550 |            1873 |           6 |           50397184 |
| 1493551813 |        2837 | 1202500801 |            28290 |             661 |           6 |           50397184 |
| 1493552138 |       56841 | 1202500801 |             2520 |             836 |           6 |           50397184 |
| 1493552728 |       54627 | 1202500801 |            14090 |             941 |           6 |           50397184 |
| 1493556803 |       20616 | 1202500801 |            14680 |             776 |           6 |           50397184 |
| 1493549781 |        1300 | 1202500801 |            20330 |            1488 |           6 |           50397184 |
| 1493556387 |       56271 | 1202500801 |            10300 |             774 |           6 |           50397184 |
```

Fig. 5 (*continued*)

Table 1
Traffic requirements versus storage requirements.

Anonymous system	Number of users	Overall traffic generated (Mbits/s)	Average network traffic per user (Kbits/s)	Storage requirement estimate (Gbytes/day)
Tor	100,000	960	9.8	14.505
Our Experiment	2956	132.36	45.85	2.0

Table 2
Average traffic incoming to the centralized logging system.

Anonymous system	Number of nodes	Throughput per node	Summary log sent to central logging system per node	Total summary log incoming to central logging system
New Model at 960 Mbits/s	900 exit nodes	1.06 Mbits/s	1.537 Kbits/s	1383.3 Kbits/s

both contents to one another. In fact the centralized system will be responsible for performing and synchronizing this task with different exit nodes on the system for two reasons: the first being simplicity, and the second for keeping track of what exit nodes may be monitored or is being subjected to an attack. The centralized system can then instruct the exit node (under attack) to no longer carry out any requests for users and then redirects these requests to other trusted exit nodes on the system. The centralized system can then perform different tests and randomize requests for the exit node to mimic end-user behavior in order not to allow an adversary to notice that the ongoing attack has been detected. Of course losing an exit node in such an attack is not that critical since more traffic and load will be distributed for other nodes; however, this falls as a compromise in terms of end-user security. The comparison of payload that is performed between exit nodes is done using encrypted checksums with public and private keys of exit nodes, as well as information about the source IPs, date and time of content delivery. At least three geographically distributed exit nodes need to be involved in this comparison in order to add more certainty to the checksums. Once the main exit node receives a request by the end user (and the centralized logging system) for Internet communication, the main exit node responsible for delivering the payload automatically notifies the other participating exit nodes, that have been instructed to verify the content, of the request and these nodes also perform the same task, compares the checksum provided by the main exit node, and report it back to the centralized logging system. In the meantime, the main exit node streams back the information through different circuits or anonymous nodes back to the anonymous user. During the streaming process the data is divided into equal size cells and are padded and a checksum is inserted for every payload in each cell. Again a copy of every cell's payload is sent to the participating exit nodes that in turn streams this information through different routes back to the end user. In the end, the anonymous user receives a copy of the payload delivered by the main exit node, a copy of the complete payload checksum delivered by the other participating exit nodes as well as the main exit node, and two copies of every cell's checksum from the participating exit nodes. The user can then perform verification through a quorum by comparing what has been received with the checksums delivered by other exit nodes. It is important to note that the pathways for delivering the checksums of every cell are different and this adds more randomness and security to verify that the content being delivered is authentic.

If the user detected a certain conflict in the checksums of the cells delivered, or in the overall checksum of the payload delivered, then the user can report this incident to the centralized logging system. The report will include either an anomaly in the encrypted checksums that have been received from the participating exit nodes, or an anomaly in the overall payload that was delivered as the participating exit node will have verified the complete payload. In the former case, this means that a circuit belonging to one of the trusted exit nodes has a node that has been compromised or a network path that is intercepting and injecting traffic. The latter case indicates that the main exit node has either been compromised since the content is not identical to the other two participating exit nodes, or a false positive due to network instability/packet corruption. In all cases, every node participating in the stream of the end-user will report actions being executed either successfully or unsuccessfully. Hence, the importance of a reputation system is not only for nodes lying in the same circuits, but also for the integrity of the payload being delivered. The user can then decide either to discard the payload delivered or to inspect it if he/she wishes to. This scheme not only verifies that the content being acquired from the Internet is authentic, but also allows the user to further verify his information at his/her end in order to further assure the anonymous user that the delivered payload has not been tampered with along the way.

5. Conclusion

This paper introduced a new technique for assuring data integrity while avoiding traffic analysis and other types of similar attacks. The new technique utilizes a collaborative based approach to allow the client to validate the authenticity of the received data at his/her end by comparing different copies of the data. By refining exit node policies when traffic is generated from and to exit nodes, anonymous systems can now be trusted and accepted by Internet entities and governments around the world. Avoiding traffic analysis, and hiding the identities of users, is the aim of any anonymous system. However, since most anonymous systems rely on aging encryption technologies for which global adversaries are a capable of compromising, then the integrity of data might be at stake. As a result a new anonymous software distribution approach has been implemented. Additionally, the paper has introduced a new method for verifying data integrity by sending hashes generated by exit nodes through different circuits; and hence, different nodes on the system. Exit nodes participate in this design whereby it is only the recipient who determines whether or not the payloads being delivered to his/her end can be trusted or not. This is done through a quorum based on the hashes received. In order to test the new model, a sample portion of an existing Internet service provider's traffic has been selected and then analysis tools, using the Service Control Engine from Cisco™, have been used to test whether or not information traversing exit nodes can be gathered using a centralized database. The tests have proved to be successful; and hence, can be adopted by current or new anonymous systems. One of the key elements that worry anonymous systems researchers is QoS for the bandwidth utilized by peers on the systems and the overall network performance. Although this has been slightly commented on, more research in QoS and a bandwidth-choking approach is required while concentrating on security and functionality implications.

References

Anonymizer – Online privacy and personal VPN software <https://www.anonymizer.com/> Retrieved 17.07.14.

Bauer, K., McCoy, D., Grunwald, D., Douglas, S. & Tadayoshi, K. (2007). Low resource, routing attacks against anonymous systems. Technical Report CU-CS-1025-07. University of Colorado, USA.

Chakravarty, S. (2014). Traffic analysis attacks and defenses in low latency anonymous communication. Ph.D. Dissertation. Columbia University.

Chaum, D. (1981). Untraceable electronic mail, return addresses and digital pseudonyms. *Communications of the ACM, 24*(2), 84–90.

China bans anonymous internet messages <http://www.theepochtimes.com/n2/china-news/china-bans-anonymous-internet-messages-35019.html> Retrieved 17.07.14.

Choi, B., Xuan, D., Li, C., Bettati, R., & Zhao, W. (2000). Scalable qos guaranteed communications services in real-time applications. In *Proceedings of the IEEE international conference on distributed computing systems, (ICDCS)*.

Coy, S. P. (2008). Security implications of the choice of distributed database management system model: Relational vs. object-oriented. Technical Report. University of Maryland.

Create a SQL Server Agent Proxy <http://msdn.microsoft.com/en-us/library/ms175834.aspx> Retrieved 17.07.14.

Danezis, G., Dingeldine, R., & Mathewson, N. (2003). Mixminion: Design of a type III anonymous remailer protocol. In *Proceedings of the IEEE symposium on security and privacy* (pp. 2–13). Berkeley, USA.

Fernández Franco, L. (2012). A survey and comparison of anonymous communication systems: Anonymity and security. Master's Thesis. Universitat Oberta de Catalunya.

Freedman, M., Sit, S., Cates, J., & Morris, R. (2002). Introducing tarzan, a peer-to-peer anonymizing network layer. In *Proceedings of the first international workshop on peer-to-peer systems – IPTPS*. Cambridge, MA, USA.

Guan, Y., Fu, X., Xuan, D., Shenoy, P., Battati, R., & Zhao, W. (2001). NetCamo: Camouflaging network traffic for qos-guaranteed mission critical applications. *IEEE Transactions on Systems, Man, and Cybernetics, 31*(4).

Haraty, R., & Zantout, B. (2014). The tor data communication system. *Journal of Communications and Networks, 16*(4). ISSN 1229-2370.

Hints, A. (2002). Fingerprinting websites using traffic analysis. In R. Dingledine & P. Syverson (Eds.), *Privacy Enhancing Technologies (PET 2002)* (pp. 171–178). Springer-Verlag. LNCS 2482.

Ibrahim, S., Abuhaiba, I., & Hubboub, H. (2012). Swarm flooding attack against directed diffusion in wireless sensor networks. *International Journal of Computer Network and Information Security (IJCNIS), 4*(12), 18–30.

Introduction to the Cisco SCE 8000 10 GBE Platform <http://www.cisco.com/c/en/us/td/docs/cable/serv_exch/serv_control/broadband_app/rel38x/sce8000_10gbe_ig/sce8000_10gbe_ig/02_SCE8000_IG_Intro_to_SCE.pdf> Retrieved 17.07.14.

Jansen, R. (2012). Privacy preserving performance enhancements for anonymous communication networks. Ph.D. Dissertation. University of Minnesota.

JAP Anonymity and Privacy <http://jap.inf.tu-dresden.de/index_en.html> Retrieved 17.07.14.

McCoy, D., Bauer, K., Grunwald, D., Kohno, T., & Sicker, D. (2008). Shining light in dark places: understanding the tor network. In *Proceedings of the eighth international symposium on privacy enhancing technologies (PETS)* (pp. 63–76). Leuven, Belgium.

Murdoch, J., & Danezis, G. (2006). Low-cost traffic analysis of tor. In *Proceedings of the IEEE security and privacy symposium*.

Ornaghi, A., & Valleri, M. (2003). Man in the middle attacks demos. In *BlackHat conference*. USA.

Ries, T., Panchenko, A., State, R., & Engel, T. (2011). Comparison of low-latency anonymous communication systems: Practical usage and performance. In *Proceedings of the ninth Australasian information security conference* (Vol. 116, pp. 77–86). Darlinghurst, Australia.

Scott, C. R. (2005). Anonymous communication in organizations: Assessing use and appropriateness. *Management Communication Quarterly, 19*(2), 157. http://dx.doi.org/10.1177/0893318905279191.

Syvneron, P., Tsudik, G., Reed, M., & Landwehr, C. (2009). Towards analysis of onion routing security. In H. Federrath (Ed.), *Designing privacy enhancing technologies: Workshop on design issue in anonymity and unobservability* (pp. 96–114). Springer-Verlag. LNCS.

Whalen, S. (2001). An introduction to arp spoofing <http://packetstormsecurity.nl/papers/protocols/introtoarpspoofing.pdf>.

Zantout, B., & Haraty, R. (2011). I2P data communication system. In *Proceedings of the tenth international conference on networks* (pp. 401–409). St. Maarten, The Netherlands Antilles.

Zantout, B., & Haraty, R. (2010). A comparative study of bit torrent and netcamo data communication systems. *International Journal of Computational Intelligence and Information Security, 1*(2), 18–28.