The NetCamo Data Communication System

Ramzi A. Haraty

Department of Computer Science and Mathematics
Lebanese American University
Beirut, Lebanon
Email: rharaty@lau.edu.lb

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. Critical information traversing networks is usually encrypted. In order to conceal the sender's identity, different implementations have proven successful - one of which is the invention of anonymous networks. This paper presents a thorough study of NetCamo - one of the most common and existing techniques used during data communication for avoiding traffic analysis as well as assuring data integrity. The paper discusses its implementation and techniques in details. The paper also presents the benefits and drawbacks of NetCamo.

Keywords: Anonymous networks, NetCamo, and data communication systems.

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, which is the Internet.

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 with where the reporter is residing in. If the reporter's identity is revealed then the reporter might be convicted. 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. Anonymous networks go beyond transferring information over the Internet, whereby theoretically, the implementations can be replicated on different communication technologies such as mobile devices, wireless networks, etc.

Before describing the details of Bit Torrent, it is important to mention that many implementations were able to achieve anonymity of the sender and receiver with some drawbacks or at a certain cost for which these implementations could, to a certain, extent prevent against traffic analysis. Anonymizer [1], JAP [2], Miximinion [3], Tarzan [4], Morphmix [5], I2P [6], TOR [7] and Bit Torrent [8] are examples of such solutions offered at the time NetCamo was being utilized.

This paper investigates the implementation of NetCamo, which is widely used today and has made a major impact on the world of networking and particularly peer-to-peer communication [9-12]. The remainder of the paper is organized as follows: Section 2 describes the NetCamo system. Section 3 presents the critique, outlining NetCamo's features, advantages as well as its drawbacks. Section 4 provides a conclusion.

2 NetCamo

NetCamo, which stands for Network Camouflage, was first introduced in 1999 by team of academic researchers in the Department of Computer Science of the Texas A&M University [13]. The team, led by Yong Guan, has forecasted that soon all traffic generated by hosts on any network and on the Internet will become encrypted at some point in time in the future. Encryption of some critical services on the Internet has already taken place whereby, for example, communication for all monetary and credit card transactions, whether using the World Wide Web or specialized protocol between hosts, are already using various types of encryption techniques. However, Guan et al. realized that encryption of such traffic and transmitting this data over the Internet or any network medium is not secure enough when subjected to special types of attacks - traffic analysis [14] happens to be one of them.

Throughout their work, Guan et al. not only present for encrypted traffic transmitted over the Internet, but also the fact that traffic is transmitted from one host to the other through a single path or route. They presented a general approach, design, implementation, as well as an evaluation of a new NetCamo system that ensures both security and efficiency for real time systems using mission critical applications while

avoiding traffic analysis. While realizing that traffic analysis is a serious security threat, that with time, an observer can develop information about the underlying systems generating this encrypted data and realize sensitive information about the type or sometimes the content of the encrypted data; with or without using cryptanalysis methods. Hence, NetCamo aims at preventing traffic analysis through two requirements:

1. Traffic Padding: Data that has been already encrypted is padded with extra meaningless data to *camouflage* the real traffic stream.
2. Traffic Re-Routing: Since data is usually transmitted from one host to the other through a shortest path or route, usually determined by the Internet or network provider, this stream of data is subject to traffic analysis by any entity observing this route. Therefore, data needs to be transmitted through different routes to reach the required destination.

The research team has predicted, tested, and solved issues related to packet padding overhead as well as traffic route planning since these aspects usually cause considerable delays in any QoS implementation. Their research has shown efficient and effective measurements when applying their system and comparing it to the performance of a non-NetCamo system with a minimal lack in performance for real system and application critical scenarios tested in private labs [15]. They described the behavior of NetCamo by defining a network and traffic model consisting of host systems and network devices such as routers and switches. At any given time these devices can be represented by a fully connected graph $G = (V, E)$ with vertices V and a set of edges E corresponding to the set of paths that may be used from a source "s" to a destination "t". The researchers argue that any observer can monitor the link (edge E) between any source and destination; and can therefore, implement a traffic analysis attack on this link; based on this scenario, traffic re-routing needs to be implemented [16].

While utilizing a QoS connection-oriented model to provide communication amongst different nodes on the network, researchers have identified two different architectures for which a connection-oriented service uses and that will affect NetCamo:

1. Integrated Service Architecture: is able to allow different connections to have different QoS parameters whereby the necessary information of each connection is kept in routers in each path, for admission control and packet forwarding purposes. Delays for all connections can be calculated based on parameters provided by existing connections and new connections incoming to the network. While the main advantage is flexibility in QoS specification, a considerable disadvantage arises when large number of connections is present and that requires a huge network with enormous generated traffic whereby router processing resources can be depleted and delay-calculations will take longer to achieve.
2. Differentiated Service Architecture: requires a configuration of pre-defined QoS classes whereby new connections will automatically be allocated into these classes. Accordingly, several connections maybe classified and served from the same class and while this reduces the QoS flexibility it also reduces high resource overhead on the network devices while accommodating large scale traffic networks. Moreover, delay-computation is minimal since each class is allocated a traffic bandwidth; and therefore, new connections can be assured or denied a service, based on resource availability.

The traffic being generated between participating hosts is presented in the form of three matrices called the payload traffic pattern matrix, payload traffic rate matrix, and the camouflaged traffic rate matrix. The payload traffic pattern matrix presents a list of hosts communicating or wishing to communicate with each other while the payload traffic rate matrix presents the rate of traffic on each edge of the graph. The camouflaged traffic rate matrix is simply the rate of camouflaged traffic sent on a stream.

In order to avoid traffic analysis through a certain edge in Graph $G = (V, E)$, NetCamo has a distorted, constant, or random traffic pattern called the camouflaged traffic pattern (CTP) whereby this pattern is implemented in a periodic process. Real stream data, that is encrypted, is actually intermitted with camouflaged traffic being inject into the live stream and sent to the destination host. The destination host, of course, recognizes the camouflaged data and drops them as soon as they are recognized as camouflaged packets.

Since a working model with QoS considerations needs to be guaranteed in mission critical systems with real-time requirements, NetCamo employed three requirements that are implemented at three different intervals. These are:

1. System Configuration Phase: This is carried out at the very beginning and before communication. This phase is composed of two steps:
    a. Determining a camouflaged pattern: a traffic pattern will be computed in order to be complex enough to be accepted as camouflaged data similar to the real data stream as well as simple enough not to cause traffic delay due to its payload since it could affect the QoS performance.
    b. Delay analysis between hosts: since vertices or hosts present in graph $G = (V, E)$ are recorded with exact edge weights, and by using a first come first serve methodology, NetCamo is able to predict the delay between hosts using any well-known packet scheduling algorithm.
2. Admission Control Phase: this phase determines whether incoming new communication streams are accepted or rejected depending on:
    a. Host-based Re-routing: a direct path from source to destination is selected that does not violate any of the system configuration phase components. If the traffic can be delivered at a rate that is in "real time" and is therefore acceptable, then the connection is accepted; otherwise, the connection is rejected. As much as this sounds as a strict rule or limitation, if a direct path is not available, the algorithm is able to exploit a number of paths at the same instance of time in order to stream

traffic while still abiding by the rules in the System Configuration Phase. This is achieved through splitting the number of packets sent into two or more different hosts that have enough resources to deliver the stream to its destination. Figure 1 below better explains how a system may deliver such a stream by considering a fully connected graph with four vertices and six edges with equal weights. Host **A** has been submitted with a stream of data to be delivered to host **B** whereby the required bandwidth is 4 Mb/sec and link capacity is 3 Mb/sec on any edge of the graph (as shown in Figure 1). Accordingly, the direct path from A to B will be rejected at first then rerouted by sending 3Mb/sec from **A** to **B** and 1 Mb/sec from **A** to **C** and then from **C** to **B** whereby the stream passing through **C** is not delayed and will arrive by the time all the traffic streamed from A to B has been delivered.

The system is also able to support multiple paths and can therefore utilize host **D** in the above example. The choice of introducing multiple paths and therefore multiple hosts is done in the next step.

b. Traffic Planning: is a complex step whereby a number of paths and hosts and/or network devices need to be accommodated in a plan that can provide avoidance of traffic analysis while not affecting the performance of the stream. Additionally, since the system caters for camouflaged traffic patterns that do not affect the time requirements of the communication, a correct traffic pattern needs to be chosen to satisfy constraints such as bandwidth stabilization, conservation, and delay. The former ensures that existing connections' payload traffic can be sent according to their traffic plan without real traffic pattern exceeding the camouflaged traffic pattern. The conservation constraint guarantees that the correct amount of traffic is rerouted through each network node or host, whereby the amount of incoming and outgoing payload traffic is equal. The latter makes sure that real time requirements are met through calculating estimations of resource availability and traffic payload and camouflaging overhead, according to the status of the network. This is done through delay computations over all direct paths and then determining the set of routes available for a stream of data to reach its host. As a result, the paths with least delays are chosen that could be a combination of direct and indirect paths.

c. Traffic Plan Generation Algorithm: Now that the above requirements have been met or computed, the system now utilizes a new algorithm that is able to select the appropriate paths for which transmission of the stream is accomplished through. The algorithm is as follows:

Given:

1. Let *s* and *t* denote the source and destination of the new connection
2. Connection QoS requirements: bandwidth P^{new} and deadline D^{new}
3. AC_{uv} is the available capacity of the direct-path from host *u* to *v*
4. $CAPp^{k}_{st}$ is the capacity of the host based rerouting path p^{k}_{st} where $CAPp^{k}_{st}$ = min p^{k}_{st} through <*u,v*> {AC_{uv}}
5. $d^{W}{}_{p}k_{st}C$ is the worst delay along path p^{k}_{st}
6. *k* is the total number of direct and indirect paths

Algorithm:

1. Step 1: for all direct or indirect paths from *s* to *t*, select a path p^{k}_{st} with the smallest delay whereby the capacity of the host-based routing $APp^{k}_{st} \geq 0$ and according to the shortest path first (OSPF)
2. Step 2: if there is no path then reject the new incoming connection
3. Step 3: if the real-time deadline cannot be met, then drop the connection
4. Else:

a. Assign q^{k}_{st} to be the minimum of either P^{new} or $CAPp^{k}_{st}$ such that $CAPp^{k}_{st}$ satisfies the QoS requirements

b. Assign now P^{new} to be the maximum of either 0 or $P^{new} - q^{k}_{st}$

c. If $P^{new} = 0$ then

- $AC_{uv} = AC_{uv} - q^{k}_{st}$ for all <u,v> which the path p^{k}_{st} passes through
- Accept the new connection

Else goto Step 1

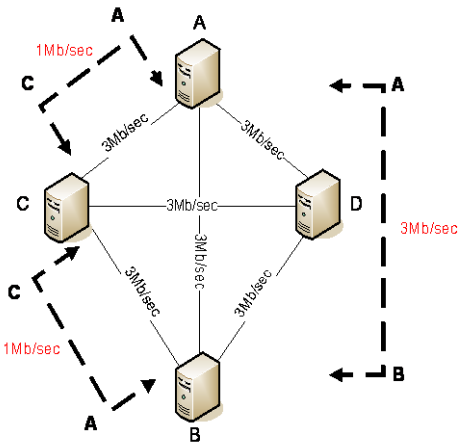

Fig. 1. Multiple data path with associated weights.

The above algorithm starts in Step 1 by simply selecting the path with the smallest delay from source to destination using direct and indirect paths as connection candidates, such that the selected path from s to t is not saturated. In steps

2 and 3 the selected path is either accepted or rejected based on the criteria of whether or not there is an available path from s to t, and the selected path meets the deadline requirements for the communication. Step 4, is basically the most important step whereby it starts by setting the minimum bandwidth requirement q^k_{st} of the connection to either P^{new} or $CAPp^k_{st}$. If $CAPp^k_{st}$ is smaller than P^{new} then another new path needs to be selected again by going back to Step 1. However, if the new path satisfies the requirements of P^{new} then the bandwidth is deducted from the class of the direct or indirect path throughout <u,v> where p^k_{st} passes through such that $AC_{uv} = AC_{uv} - q^k_{st}$. The process of selecting new paths is done again until there are no new paths available or that meet the requirements of the new connection. What is important to note that P^{new} is the bandwidth requirement for a new connection N; therefore, if communication or a stream of data needs to traverse the NetCamo network then the bandwidth requirement P^{new} needs to be pre-allocated at every path or indirect path along from source *s* to destination *t* prior to communication. Moreover, it is unclear whether or not P^{new} changes dynamically throughout the algorithm's execution whereby host-based rerouting occurs and therefore the total rate P^{new} is now decreased from the original required value since now the stream of traffic is going to be split into different routes. Going back to the Admission Control Phase explains that this can indeed happen since host-based rerouting is capable of splitting data through direct and indirect multiple routes satisfying the real-time and QoS requirements of the communication.

3. Run-time Phase:
During this phase traffic passing through the network is now intermitted with dummy packets whereby meaningless packets are injected into the stream to camouflage the real traffic. The objective of the padding process is to make the real traffic pattern match the camouflaged traffic pattern as much as possible. The algorithm is based on a time interval basis whereby dummy packets are injected at a period of time equal to 1/α. This α depends on the packet payload and the delay and other factors that affect the network. The run time phase applies low-level traffic control to compensate for fluctuations during communication on all paths in order for the stream to appear uniform and constant from source "*s*" to destination "*t*".

The NetCamo system has four separate entities. The first and the second are two hosts, the third is the NetCamo Traffic Controller, and last is the network. One can obviously realize the number of sub-components in each entity:

a. The NetCamo Network Controller contains a single NetCamo traffic manager and two router and host agents depending on the number of hosts present on the network. The traffic manager is responsible for delay calculations, real-time connection admissions control and traffic planning. The router agents control and monitor routers on the network used in the NetCamo system while the host agents accept the connection admission and forwards new connections to the NetCamo traffic manager, which in return communicates with the NetCamo host controller to control the rate of traffic and host-based rerouting policies.

b. The NetCamo Host Controller, whilst residing in each host, contains two components; the host manager and the traffic controller. The traffic controller is responsible for rerouting and sending dummy packets to other hosts. Whereas the host manager forwards the traffic plan information from the NetCamo Network Controller to the traffic controller in order to controls as well as monitor the traffic and status of each host.

c. Applications are able to use a NetCamo API to establish or destroy connections. The entire implementation of NetCamo has been carried out on the kernel level whereby to the left of Figure 3. illustrates the location of the NetCamo's Traffic Controller and Host Manager on a Windows NT implementation.

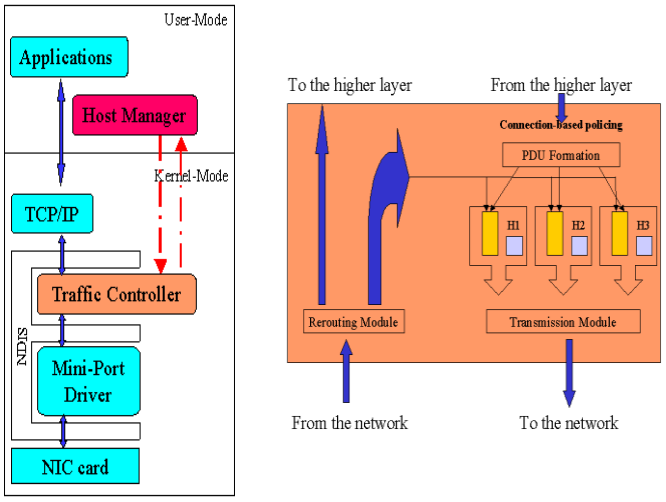

Fig. 2. Implementation and location of the NetCamo's traffic controller.

Figure 2 shows that for every host residing on the network and that is able to transmit and receive information, there is a queue at each host and for each host that is sending and receiving information using host-based routing mechanism. Moreover, the routers that reside in this network contain queues for each participating entity as well.

3 CRITIQUE

The NetCamo system is indeed one of the new and brightest ideas that addresses securing a stream of traffic from source to destination while still abiding by QoS specifications defined by that stream. The system is able to distribute traffic through

different predetermined paths and routes in order to prevent against traffic analysis and traffic sniffing attacks using a camouflaged algorithm. However, there are a couple of points that need to be mentioned about this system:

1. Since this research and implementation has been accomplished for real time systems with specific QoS qualifications that cannot be altered, then it is expected that NetCamo be a reliable system that is able to accept communication 100% of the time and to move that traffic from point A to B through different routes. NetCamo at the moment can only accept new incoming requests 85% of the time due to many limitations in the distributed system. One can imagine NetCamo as acceptable if for instance a new demanding QoS communication connection can wait until it is admitted into the system, then when NetCamo can assure real-time delivery it is therefore admitted. However, in most scenarios there will be a large number of connections and the 85% value may decrease considerable as the number of incoming connections increases. Accordingly, more than 15% of incoming connections will be rejected (not even queued) until the network QoS requirements are available for the new incoming connection.
   One can also imagine a scenario whereby a system may be saturated with many minimal QoS-requirement connections and a single demanding new connection may wait forever while other minimal QoS requirement connections are being admitted into the system. During this saturation phase the system may not even queue and then free resources to admit this connection which will always be dropped.
2. NetCamo depends on hosts for host-based rerouting, and routers for direct path traffic, to make the NetCamo system efficient and effective. However, during all tests, geographical distribution of hosts and routers was not considered. This is really vital if it needs to be implemented in the real world.
   Imagine a scenario where the NetCamo system is implemented in North America and then a source in North America needs to send traffic to a host in Hawaii. Traffic will therefore bounce through different routes and hosts based on the NetCamo algorithm and then will be transmitted through the same path or link to the host Hawaii from the same or different network hosts; however, using the same number of predetermined physical connections linking to Hawaii. The NetCamo system would have accomplished in this case only data camouflaging, since the host in Hawaii should be part of the same NetCamo system. This scenario could have been implemented using data encryption whereby camouflaging adds an additional layer of security to the transmitted stream of traffic. However, traffic analysis and traffic sniffing can still take place in this scenario.
3. The NetCamo system is dependent on hosts and routers in the sense that a single managing component needs to take control of all router and hosts in the system in order to determine whether new communication can be admitted as well as control and monitor traffic in the NetCamo system. In the real world, hosts can only be controlled if they wish to join the NetCamo system. However, routers cannot be controlled unless they are privately owned, dedicated to the NetCamo network, and located at core networks strategically, so that they have enough bandwidth and traffic capabilities to talk to all hosts and control the traffic rate of all hosts. This may indeed be a difficult task to implement in the real world.
   Additionally, the NetCamo managing-component must control the queues and rerouting of traffic in order to assure that new QoS real time system connections arrive at their destination correctly. As the number of connections increases or if a demanding connection is rerouted through a host with bandwidth X and the traffic requirement is close to X (or X itself), then the host becomes dedicated for this communication and is rendered useless until the communication is over. In the real world, a host may be a workstation that is heavily used and depended-on for work related tasks. Therefore, hosts that may join the NetCamo system need to have enough bandwidth and may be active during idle times and not during working hours.
4. The NetCamo system is dependent on routers to set the rate of traffic and control the network flow while gathering statistics about delay and other related network parameters. If routers become utilized for all host-to-host communication then traffic passing through these routers may be detected and if one of the routers is compromised then the NetCamo traffic passing through this router may be analyzed.
5. The NetCamo system has been tested in a LAN environment with latency between network components equal to 20msec. The system needs also to be tested at different and remote geographical locations since latency may add more error to the delay computation algorithm as the number of admitted and processed connections running through the system increases drastically.
6. NetCamo utilizes a centralized management node to control different functionalities and variables in the system. Its implementers have suggested a decentralized implementation that one finds extremely necessary in order to consider this system reliable and effective. Moreover, if decentralization is not considered in upcoming future implementations of NetCamo then attacks on the management node can also render the system to be useless in case of a directed attack on the management node.
7. Since traffic padding is applied at a rate equal to $1/\alpha$ then with time and an enough amount of traffic and traffic analysis tools, one can determine α, if the rate is kept at a constant. The system must therefore randomize α without affecting the camouflaged algorithm and other components in the system.
8. The algorithm for choosing paths from source to destination, considers all available routes including

direct and indirect paths. However, what happens if a host, for which a connection or multiple connections is dependent on for transmitting a data stream, fails during the transmitting phase? One needs to mention that these hosts need to be reliable and can assure the amount of bandwidth they have been allocated. Moreover, the system should also have different routing plans in sudden failures during data streaming, so that data is redirected to other hosts immediately once a failure has been detected. Hence, the failure of a single route or host in NetCamo could mean that the system has failed in assuring the communication the QoS specifications it has guaranteed to assure.

9. To be able to receive/send data using NetCamo, a host needs to be part of the NetCamo system. Therefore, the host needs to announce that it has a certain amount of bandwidth to be able to send and receive and communicate with the NetCamo management system and other hosts on the network. The problem arises when an un-trusted host joins the NetCamo system and then it announces that it has more bandwidth then it actually has. The NetCamo system will then send, and expect to receive at a much higher rate and the un-trusted host may maliciously drop packets, or interfere in malicious ways to make the NetCamo system unreliable and unable to meet a connection's QoS requirements. Now imagine a swarm attack [17] of these un-trusted hosts or hosts that are compromised. The NetCamo system will fail in assuring its connections' proper communication, if not communication at all.
10. The NetCamo system must therefore not consider accepting any host as a trusted in the NetCamo network. Still, any host wishing to send and receive data using the NetCamo system should be able to do so easily.

    A workaround for this could be that any host wishing to use NetCamo can assign one of the NetCamo hosts as a proxy for its communication. It would then redirect the traffic it needs to be secured through this host to send and receive data. Once data communication needs to be established, the NetCamo host would authenticate the new host communicating with it using a predefined username and password before accepting communication. The NetCamo host will then call the NetCamo management node and inform it about the QoS requirement of the candidate stream that is pending admission. Once admitted, data being sent from each node and router in the NetCamo system is camouflaged; however, data is not camouflaged between the new host and the first contacted NetCamo host.
11. NetCamo claims end-to-end prevention for data-sniffing and traffic analysis. Frankly, one finds this impossible when it comes to attacks occurring on the LAN level, such as man-in-the-middle attacks [18] and ARP poisoning with many-to-many targets [19], unless the network infrastructure is secured against these threats which not all organizations have implemented as a standard [20-22].

4. CONCLUSION

This paper presented the NetCamo anonymous system and its corresponding details that have made such a system a success. 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. One of the key elements that worry anonymous systems researchers is quality of service for the bandwidth utilized by peers on the systems and the overall network performance. Although this has been slightly commented on, more research in quality of service and a bandwidth-choking approach is required while concentrating on security and functionality implications [23-24].

Acknowledgments. This work was funded by the Lebanese American University – Beirut, Lebanon.

References

1. JAP Anonymity and Privacy, http://jap.inf.tudresden.de/index_en.html.

2. Danezis, G., Dingeldine, R. and Mathewson, N.: Mixminion: Design of a Type III Anonymous Remailer Protocol, Proceedings of the IEEE Symposium on Security and Privacy, Berkeley, USA, (**2003**), pp. 2-13.

3. Freedman, M., Sit, S., Cates, J. and Morris, R.: Introducing Tarzan, A Peer-to-Peer Anonymizing Network Layer, Proceedings of the First International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, (**2002**).

4. Rennhard, M and Plattner, B.: Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection, Proc. of the ACM Workshop on Privacy in the Electronic Society, Washington, (**2002**).

5. The Anonymizer: Frequently Asked Questions, http://anonymizer.com.

6. Zantout, B. and Haraty, R.: I2P Data Communication System, Proceedings of the Tenth International Conference on Networks, (**2011**), pp. 401-409.

7. Haraty, R. and Zantout B.: The TOR Data Communication System. Journal of Communications and Networks. ISSN 1229-2370, (**2014**).

8. Zantout, B., and Haraty, R.: A Comparative Study of BitTorrent and NetCamo Data Communication Systems. International Journal of Computational Intelligence and Information Security, (**2010**) Vol., Issue 2, pp. 18-28.

9. Singh, R., Singh, P., Duhan, M. An Effective Implementation of Security based Algorithmic approach in mobile adhoc networks**,** *Human-centric Computing and Information Sciences* (**2014**), 4:7.

10. R.Sumathi and M.G.Srinivas, A Survey of QoS Based Routing Protocols for Wireless Sensor Networks, Journal of Information Processing Systems, (**2012**), Vol. 8, No. 4.

11. Samer Moein, Fayez Gebali, Issa Traore, Analysis of Covert Hardware Attacks, Journal of Convergence (**2014**), Volume 5, Number 3.

12. Youngseok Chung, Seokjin Choi, Dongho Won, Lightweight anonymous authentication scheme with unlinkability in global mobility networks, Journal of Convergence, (**2013**), Volume 4, Number 4.

13. Guan, Y., Fu, X., Xuan, D., Shenoy, P., Battati, R., and Zhao, W.: NetCamo: Camouflaging Network Traffic for QoS-Guaranteed Mission Critical Applications, IEEE Trans. on Systems, Man, and Cybernetics, (**2001**), 31(4).

14. Danezis, G. and Clayton, R.: Introducing traffic analysis. In Digital Privacy: Theory, Technologies, and Practices, A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. di Vimercati, Eds. Auerbach Publications, Boca

Raton, FL, (**2007**), Chapter 5, pp. 95–117.

15. Choi, B., Xuan, D., Li, C., Bettati, R., and Zhao, W.: Scalable QoS Guaranteed Communications Services in Real-Time Applications, Proceedings of the IEEE International Conference on Distributed Computing Systems, (ICDCS), (**2000**).

16. Choi, B., Xuan, D., Li, C., Bettati, R., and Zhao, W.: Efficient Traffic Camouflaging in Mission-Critical QoS Guaranteed Networks, Proceedings of the IEEE Information Assurance and Security Workshop, West Point, Virginia, USA, (**2000**), pp. 143-149.

17. Ibrahim S., Abuhaiba, I., and Hubboub, H.: Swarm Flooding Attack against Directed Diffusion in Wireless Sensor Networks, International Journal of Computer Network and Information Security (IJCNIS), (**2012**), Vol. 4, No. 12, pp. 18 – 30.

18. Ornaghi, A., and Valleri, M.: Man in the Middle Attacks Demos. In: BlackHat Conference, USA (**2003**).

19. Whalen, S.: An Introduction to ARP Spoofing, http://packetstormsecurity.nl/papers/protocols/introtoarpspoofing.pdf, (**2001**).

20.. Madhu J Sharma, Victor CM Leung, IP Multimedia subsystem authentication protocol in LTE-heterogeneous networks**,** *Human-centric Computing and Information Sciences* (**2012**), 2:16.

21. Fan-Hsun Tseng, Li-Der Chou, Han-Chieh Chao, A survey of black hole attacks in wireless mobile ad hoc networks**,** *Human-centric Computing and Information Sciences* (**2011**), 1:4.

22. Fikadu B. Degefa, Dongho Won, Extended Key Management Scheme for Dynamic Group in Multi-cast Communication, Journal of Convergence (**2013**), Volume 4, Number 4, pp. 7-13.

23. Wei Nie, Houjun Wang and Jong Hyuk Park, Packet Scheduling with QoS and Fairness for Downlink Traffic in WiMAX Networks, Journal of Information Processing Systems, (**2011**), Vol. 7, No. 2.

24. Sunil Kumar, Mahasweta Sarkar, Supraja Gurajala and John D. Matyjas, MMMP: A MAC Protocol to Ensure QoS for Multimedia Traffic over Multi-hop Ad Hoc Networks, Journal of Information Processing Systems, (**2008**), Vol. 4, No. 2.