

Article

Not peer-reviewed version

Cryptographically Upgrading TOR Network to Enforce Anonymity by Enhancing Security and Improving Performances

[mohamed chahine ghanem](#) *

Posted Date: 14 July 2023

doi: 10.20944/preprints202307.0982.v1

Keywords: TOR; Online Anonymity; Onion Routing; Multi-layer encryption; authenticated- encryption; ExperimenTOR; JAVA Crypto; Circuit Construction; Routing protocols; Cell Encapsulation; AES; OCB



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Cryptographically Upgrading TOR Network to Enforce Anonymity by Enhancing Security and Improving Performances

Mohamed Chahine Ghanem

affiliation

Abstract: The Onion Route Network (also called TOR) is by far the most efficient and widely used anonymity platform with millions of users daily and an expanding size and capacities. Since its public deployment in 2002, the Onion Routing network (also known as TOR) has maintained its leading position and dozens of propositions aiming to improve its performance and enhance the security (anonymity and privacy) have been made. Given the significance of this research area, this work seek to contribute into the improvement of TOR by investigating and testing revolutionary cryptographic and routing mechanism. This work is justified by the current TOR vulnerability and observed weaknesses, and set the challenging aim of covering these security flaws by proposing the relevant security and performances improvement such as the authenticated-encryption for onion construction, the lightened onion encapsulation approach and the secured circuit selection and cell routing mechanisms. The obtained results from implementing the proposed improvements and testing them into a TOR-like simulation platform permit to validate not only, the performances and security contributions brought by the improvement, but also the suitability of their potential implementation into the real TOR network.

Keywords: TOR; Online Anonymity; Onion Routing; Multi-layer encryption; authenticated- encryption; ExperimenTOR; JAVA Crypto; Circuit Construction; Routing protocols; Cell Encapsulation; AES; OCB

Chapter 1: Introduction

1.1. Background

When the data is transmitted over the Internet, it is routed from one part to another via the TCP/IP protocol in which the data is sent in a piecemeal manner as a series of packets, each packet is composed from a header containing routing information and a payload containing the fixed-size actual data. If a client Alice wants to confidentially exchange communication with a server or another client Bob, she will be required to encrypt the payload of each packet to conceal their content from an eventual Eve who could be eavesdropping the communication. Nevertheless, making the communication content confidential does not conceal the fact that Alice is communicating with Bob which could be as important to Eve as the content of their communication. By viewing the routing information in Alice's packets, Eve can easily see that Alice is exchanging data with Bob and therefore in this case, the encryption guaranteed the confidentiality of the data but not the communication itself.

In order for Alice to be fully protected, she must not only conceal her transmitted data but also she need to hide the fact that she is communicating with Bob by making packet anonym during their transmission to Bob. During the last decade, several dedicated anonymity systems have been developed to offer this specific protection. Anonymity system are usually categorised into two types: low latency and high latency systems. Since its public deployment in 2002, the Onion Routing network (also known as TOR) has maintained its leading position as the most efficient and effective technology for anonymous web browsing (Backes et al., 2012). Onion Routing is a distributed, low-latency and overlay network which currently employs several thousand of dedicated relays and was designed in order to allow users of TCP-based applications an online anonymity (Dingledine et al., 2014) and also to evade governments' internet monitoring and censorship.

The onion name is mainly due to the Multilayer Encryption Method used by the relays in order to provide confidentiality. In fact, the encryption mechanism was designed in order to prevent the relays from accessing to the data which they are routing and even the whole path that the data will take to reach the final destination of the packets. TOR, basically, works in this way: the Clients choose a path through the network and build a circuit, in

which each Onion Router (or simply called node) in the path knows only its predecessor and successor, (Camenisch & Lysyanskaya, 2005). The Data is firstly encrypted in several layers (the number of nodes which constitute the path) and sent to an entry node using an encrypted channel. Afterward, it will be transmitted to an exit node following the pre-planned path of the OR network (relay to relay). Finally, once the packet reaches the exit node which is the only relay which will have access to the address of the final destination and will transmit it without encryption in a clear way (Haraty & Zantout, 2014).

The implementation of the second generation of TOR didn't cover appropriately all the security issues, in fact, some TOR protocols remained vulnerable and especially TOR authentication protocol (TAP) and some others improvements were introduced just after that Zhang proved that the security of TAP does not imply the security of the overall system especially against concurrent execution attack (Zhang, 2009). In fact, as the TOR protocol security will not be guarantee by only the sequential execution of multiple TAP but also by securing the paths construction mechanism and most important the multilayers encryption and processing algorithms (Benmeziiane et al, 2011).

The most important current challenge which TOR is facing are resisting to internet traffic analysis which could lead to linking several intercepted communications to the corresponding parties or linking multiple communications to a user (Nia et al., 2014). To deal with these emerging challenges some improvements should be introduced to TOR. Securing OR modules consists mainly of two parts: firstly, securing the onion construction algorithm, and secondly, implementing a secure way for key exchange such as "the one-way authenticated key exchange protocol" (1WAKE) which was introduced for the first time, in 2011, by Goldberg et al. (Backes et al., 2012).

Previous works had already tackled the integrity issue and concluded that TOR can guarantee an end-to-end rather than hop-to-hop integrity (Backes et al., 2012). This work will focus on Onion construction algorithms which typically use several layers of symmetric encryptions proceeded and followed by operations and keys exchange protected by Public Key encryption along with the use of integrity and authenticity check mechanisms, such as Hashing and MAC.

The aims behind this work is to study formally and academically the existing onion wrapping and unwrapping algorithms over them four core properties: correctness, security of statefulness, synchronicity and cipher-text in order to identify the current weakness reasons and try to introduce some improvement to the existing encryption policy in order to enforce which will be the main topic in of this work (Kate & Goldberg, 2010).

1.2. Work aims and objectives

As TOR imposed itself as the universally uncontested online anonymous communication tool and attract more and more user across the world for both professional and personal use. Therefore, the security of this network should be formally reviewed and some components adopted by TOR to have to be improved to guarantee security and enhance performances.

The aims targeted by this work are to, firstly, study cryptographically the existing TOR's cryptosystem and improve the existing solution in order to cover all the existing vulnerabilities that are making TOR a target for different kind of attacks. Secondly, reduce the current delays that TOR is experiencing by balancing and optimizing the use of resources and especially those responsible of performing encryption/decryption and routing activities. Finally, improving the network overall security and traffic illusion over the internet by introducing a new way of defining routing circuit and also introduce the sessional multi-path routing for TOR cells to enhance security and eliminate the risk of successfully de-anonymise or link TOR traffic to one or more user.

The primary objectives of this work are related to the improvement of the current TOR cryptosystem by enforcing the existing end-to-end integrity with a node-to-node authentication checking mechanism. This approach will help to resolve the current weaknesses of data authenticity within TOR without using the heavyweight control methods proposed by Backes et al. (2012). The multi-layers encryption called "Onion Encryption" will be improved by introducing AES-OCB (Off- Set Code Book mode) which is a block cipher encryption allowing the achievement of both confidentiality (privacy) and Authenticity simultaneously, it is also a very efficient algorithm especially by allowing a perfect parallelism in operations and working in online mode. The TOR crypto-system should also be improved by introducing a new way of constructing the Onion (multi-layered TOR cells encapsulation and encryption) and exchange keys between different parties (OP, ORs, DA).

The second set of objectives targets the performance improvement and the reduction of delays caused by the high level and useless redundancy of encryptions implemented especially on Onion construction (encapsulation). Despite the fact that TOR network is considered as a "low latency", the heavy-weight encryption and the routing

and circuit construction mechanisms are still causing delays. Therefore, this issue could be sorted out by introducing a much efficient implementation of the algorithm responsible of performing multi-layered encryption which will considerably enhance the TOR performances and reduce delays.

The last set of objectives is related to the network security and resistance to the modern attacks especially regarding the construction of routing circuit, and the shipping of cells through TOR. This work will propose and test several improvements regarding the choice of ORs part of the circuit, the multi-path (circuit) routing approach and the dynamic circuit length approach. All these features will contribute to enhance TOR security and performance by guaranteeing a more security to the users against specific attacks and enhance anonymity, confusion and diffusion of TOR traffic and avoiding user exposition and recognition by their data pattern.

1.3. Contribution

TOR's status as worldwide leading solution guaranteeing users' online anonymity and privacy along with its capabilities to resist to Internet censorship has attract more enemy among the dictatorship government, hackers and even unethical researcher, whose are constantly designing and developing new technologies and techniques to attack and compromise TOR security or performance. On the other hand, TOR developer community which closely works with academic researcher around the world are committed to keep improving TOR security and performance by enhancing design, adopting new standardised techniques and solution and enlarging the use of TOR to other application such as mobile phone. Since TOR establishment, researchers have been continuously studying, criticising and proposing enhancements in term of security, anonymity, and performance in an effort to make TOR more secure and flexible to use which will, as result, attract more user and this improve the network's resilience to various attacks. Nevertheless, the researches focussed mainly on how attacking and proving TOR limits and vulnerabilities, where few works emphasised on improving TOR overall security which will obviously cover the weakness of the current design.

This work will bring an addition to TOR development community continuous work by proposing a mixture of security improving cutting-edge cryptographic solutions and performance enhancing implementation of the proposed improvements. There was always a balance of security and performance or usability in cyber security, and this work will not present an exception. The proposed improvement will be implemented in a dedicated simulation platform, tested and compared with the existing TOR deployment. The results will be deeply discussed and the approved improvements will be highlighted where the dropped proposition will constitute another research starting point.

Chapter 2: Literature Review

During this chapter, a deep research outcome will be presented regarding the online anonymity generally and TOR network especially. A review of the past and current research on TOR will be performed and the chapter is concluded by a discussion of ethical, legal, social and professional issues related to TOR research.

2.1. Online Privacy and Anonymity:

Before presenting the existing anonymity networks and platforms, it is important to introduce some key concepts and aspects related to the cyber security field which will be heavily used during this work. The importance of accurately defining our topic terminology is due to the fact that many people including academics are still confounding the two concepts of online Privacy and online Anonymity. The two concepts are in fact two different concepts, they are both beneficial for not only individual, but they are becoming a society's interest. In this section we will discuss the two important concepts.

Hiding the identity of internet users and ensuring (preserving) theirs privacy while online is a very challenging task. Initially, the Internet was designed to guarantee the connectivity and none of its initial goals included guarantying the users' privacy or anonymity. Therefore, the current internet implementation allows a third party with adequate capabilities to see who is communicating with whom and in major cases the content of the communications. Therefore, internet users' tracking and monitoring for different purposes varying from the simple targeted marketing operations to governmental large scale monitoring and censorship is becoming a fact. Nowadays, with increasing threats against human rights and individual freedom ranging from routine governments surveillance and data collection to illegal human right breach in some cases for different purposes made from internet

an untrusted mean of communication. Thus, Anonymity system became crucial tools for a variety of targeted people aiming to protect themselves when using internet.

2.1.1. Online Privacy

Privacy is a major concern or all Internet users and is becoming more difficult to set a fixed perimeter of privacy online. Internet users' privacy is controversial aspect which is still being debated by the internet community, government and NGOs. users assumptions about the control level they have control over their private information is usually wrong and the best example is when they engage in online activities such as online social networking which is essentially based upon sharing of private information but they consent of disclosing such information to known or unknown parts. Moreover, over years and the importance that constitute such information, entire organisation are devoted to compromise the user privacy for different purposes varying from legal to criminal and from business to national security.

2.1.2. Online Anonymity

Online anonymity is occur when an internet user is not identifiable (distinguishable) within a set of other users, Online Anonymity is righteous and necessary in many scenarios, such as protecting Internet user privacy, improving system security, bypassing Internet censorship, satisfying some antivirus requirement, and protecting Internet users' computer from hackers' attacks. With the wider use of the Internet and more improving computer hacker technologies, the Internet's users are anxious to gain more powerful ability to keep their privacy and security. They want to request a better network tool or technology to protect their private information from being monitored by the Internet sniffers and hackers. With this requirement, anonymous communication has become more and more popular on the Internet (Kate & Goldberg, 2010).

Anonymity is righteous and necessary in many scenarios, such as protecting Internet user privacy, improving system security, bypassing Internet censorship, satisfying some antivirus requirement, and protecting Internet users' computer from hackers' attacks. With the wider use of the Internet and more improving computer hacker technologies, the Internet's users are anxious to gain more powerful ability to keep their privacy and security. They want to request a better network tool or technology to protect their private information from being monitored by the Internet sniffers and hackers. With this requirement, anonymous communication has become more and more popular on the Internet (Kate & Goldberg, 2010).

2.2. Anonymous Communication Networks

With more and more sensitive and private communications transiting over the Internet, there was a need to develop platform and solution which can guarantee and maintain the privacy and security of these communications. (Dingledine et al., 2014). Anonymity communication networks (ACNs) emerged as a the perfect solution allowing people to conceal their identities online by guaranteeing the un-linkability of the users' IP addresses, their digital fingerprint, and their online activities (Pfitzmann & Hansen, 2008). Anonymous communication networks became an essential component in the nowadays challenging cyber security world. Overall, there two main categories of ACN; high-latency systems and low-latency systems.

High-latency systems such as Babel and Mixmaster tend to sacrifice performance to ensure a perfect security especially against type of threads called correlation attacks in which Eve monitor Alice's packets timing and patterns to perform a deep and complex statistical analysis aiming to match both part of each communication, these systems use advanced techniques such as delaying packets and injecting invalid packets in order to gather as much information they can get to produce an accurate time or pattern matching. From their part, the developer of this solution are constantly introducing heavyweight protection technologies, thus the transmission delay (latency) on this solution is high making this solution unsuitable for sensitive transmission activities such as HTTPS in which the time accuracy is crucial. High-latency anonymity networks can only guarantee the anonymity for applications which can tolerate an intentional delays such as e-mails and blogs (Danezis et al. 2003).

Nevertheless, as the major part of nowadays online activities is interactive such as browsing the web, instant messaging and social media, the high-latency system were useless and didn't offer the intended flexibility of use. Therefore, low-latency anonymity systems have been developed to cover this issue. The low-latency systems attempt to balance the security (confidentiality, integrity, availability, anonymity and privacy) and usability

(flexibility) by avoiding causing long delays on the data transiting through these systems despite the fact that the overall security will present some flaws and be exposed to different type of advanced attacks. As one of world leader and most famous low- latency anonymous communication system, TOR is getting more and more trusted and used around the world. Many organisation and government, which pay particular attention to their members and employees' online activities exposure, advise their staff to use TOR for their professional online activities. Moreover, some organisations and even enforcement agencies recommend that their employees use TOR when communicating with others on the Internet (Darcie et al., 2013)

2.3. TOR Network

2.3.1. History of TOR

TOR was originally designed and prototyped by Sun Solaris and have been since implemented as part of web browsing application in order to provide secure remote login and sanitising users' data while transmitting it through internet. The idea of Onion Routing was mainly inspired from the work of David Chaum (Haraty & Zantout, 2014). Nevertheless, several enhancements were introduced by F. Syverson, D. Goldschlag and M. Reeds when the project was adopted, In 1995, by the US army Naval Research Laboratory (Syverson et al., 1997) as the future anonymous communication platform allowing MoD staff to use Internet while remaining anonymous alongside with preventing enemies from performing traffic analysis and eavesdropping of the associated traffic. The Onion Routing (TOR) took place as the name of the platform and research went to life and started the implementation of almost hundred router (node) placed in different locations (corporations, government departments and offices, and academic institutions) and the project was allocated a significant attention of internet security community within the MoD and US government in general.

In 1997, the project received official funding from the famous US Department of Defence Advanced Research Projects Agency (DARPA) and was classified as a High Confidence Program. Furthermore, improvement efforts and research works were introduced to the initial implementation especially the routing and encryption algorithm and the network was running with an average 50,000 use per day and reached a peak of 84,022 simultaneous connections at the end of 1998 (Haraty et al., 2014).

During the period between 1999 and 2002, the project suffered from the absence of funding and support and was abandoned on 2002. Nevertheless, in 2004, and after the project was resumed by a group of volunteers named just after TOR project, TOR won the Edison Invention Award, and shortly after the first generation of the code was officially dropped to leave the place to the second-generation onion routing. To this date, the project are funded by volunteers across the world and users' donation and it is still under development and constitute the largest network testing platform in the world relying on almost six thousand dedicated routers located worldwide and generating a bandwidth capacity of almost 200 Gb/s (Figure 1) of bidirectional data streams (Dingledine et al., 2014).

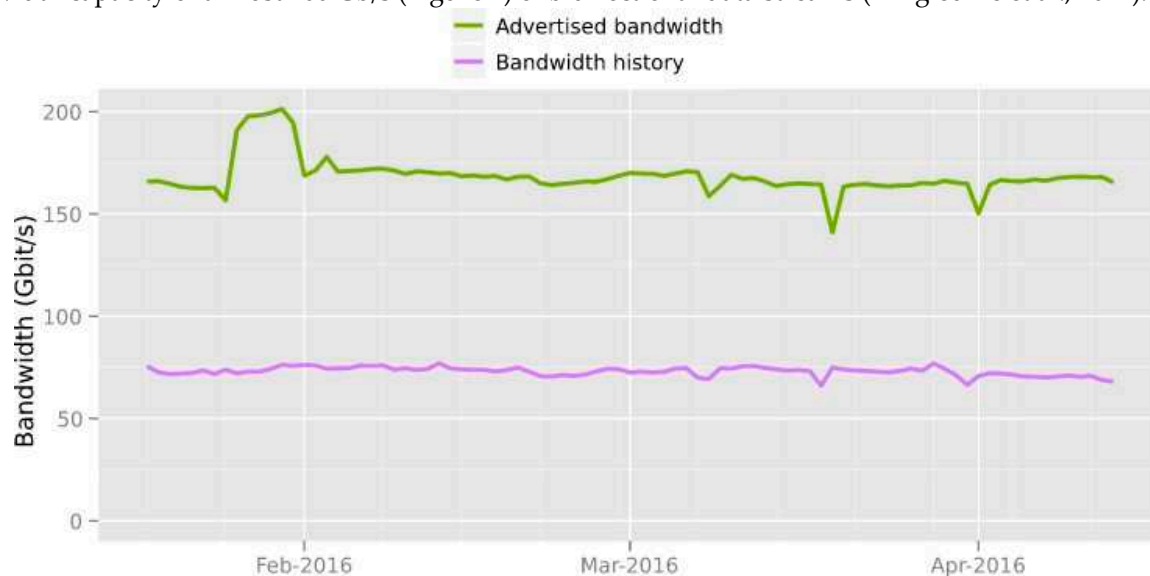


Figure 1. The current TOR bandwidth capacity and evolution across time (TOR Project, 2016).

Later in 2012, a third-generation of TOR was deployed and took the place of the first one, several new functionalities were added in order to maintain the efficiency of the solution which faced hard time especially with some weaknesses revealed due to the exploits against MD5 hashing which was replaced by SHA256, DES symmetric encryption which was replaced by AES128 (Danezis et al., 2010). The sudden increase of the computing power and parallel processing was also a big issue for TOR's cryptanalysis resistance. The new version was improved by adding a revolutionary protocols and functions such as: the new Forward Secrecy protocol, new shipping protocol in which several TCP packets can share the same circuit, introduce Leaky-pipe circuit topology and a new Congestion control, new dynamic exit policies, source-to-destination integrity checking and hiding the used services. Some existing function were removed such as mixing, padding, and traffic shaping where some others were modified like the Separation of "protocol cleaning" from anonymity protocol (Dingledine et al., 2014). In UK, it is estimated that a couple of thousands of users are using TOR on regular basis (Figure 2) for different purposes varying from legitimate protection (activist, politics and journalist) to illegal activities (cyber-criminals, terrorists, hackers).

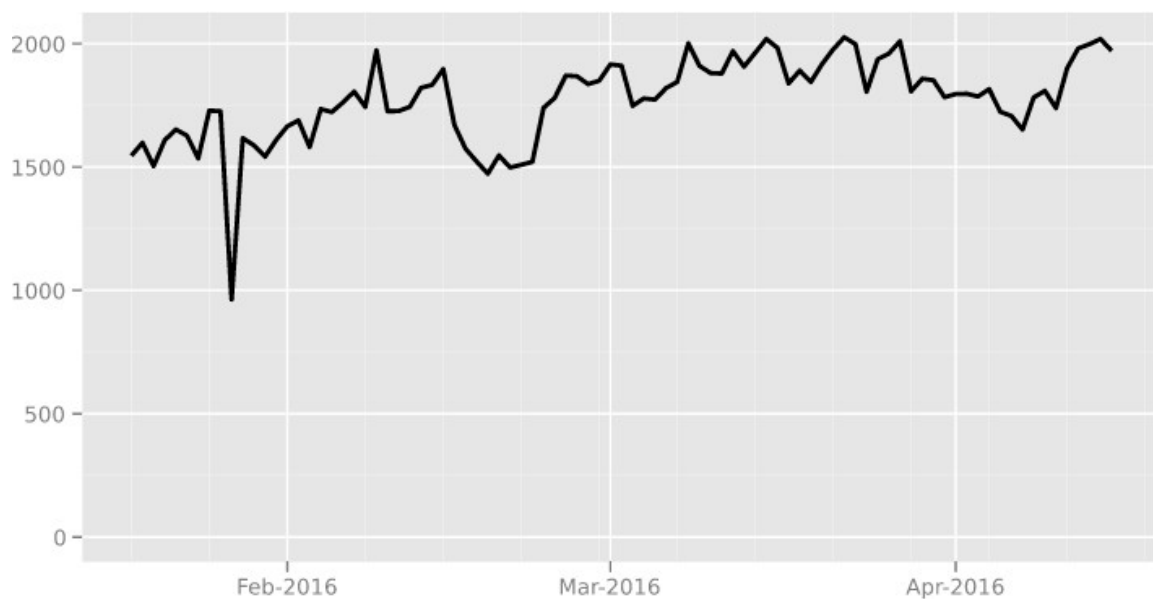


Figure 2. the number of regular TOR user in the UK for the first quarter of 2016 (TOR Project, 2016).

2.3.2. About TOR:

TOR network is the most famous and widely used low-latency anonymity platform, it relies on multi-layer encryption in form of onion and special routing concepts (Reed et al., 1998) to achieve anonymity. Today, the network consists of about 6000 routers operated (run) by volunteers across the world (Tor Project, 2016) which are called Onion Routers. Each OR is a part of one or more routing path called Circuits and identified by its contact information (IP address, ID, public keys, geographic location and other functional parameters). All these information is stored into an independent directory authorities (DA) which is responsible of the network consensus and control the network with a minimum information requirement. Onion Proxy (OP) downloads from the DA the consensus documents and the descriptors and use it to establish communication circuits through TOR network before to reach their Internet destinations. Currently, each TOR circuit is composed from three nodes (ORs): Entry guard OR, Middle OR, and Exit OR and traffic is transiting throughout the network in form of fixed-size unites called Cells.

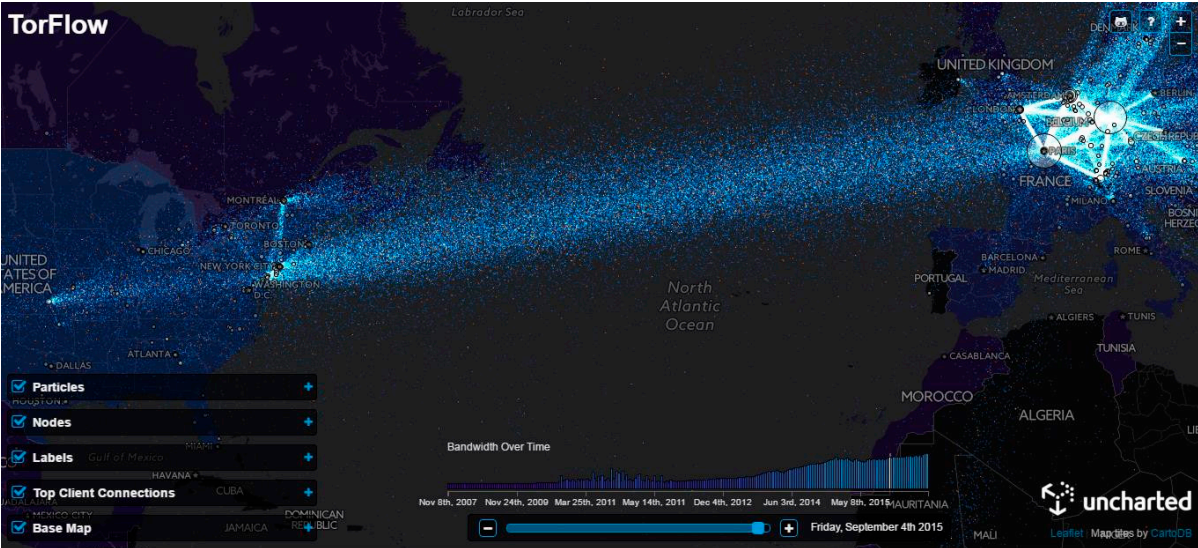


Figure 3. TOR traffic and activity across the world (TOR Flow, 2016).

The following graph shows the number of running ORs that have had certain flags assigned by the directory authorities. These flags indicate that a relay should be preferred for either Entry (Guard) or Exit positions. Moreover it show the status of the relays especially those with high-bandwidth (Fast) or long-lived (Stable or Trusted).

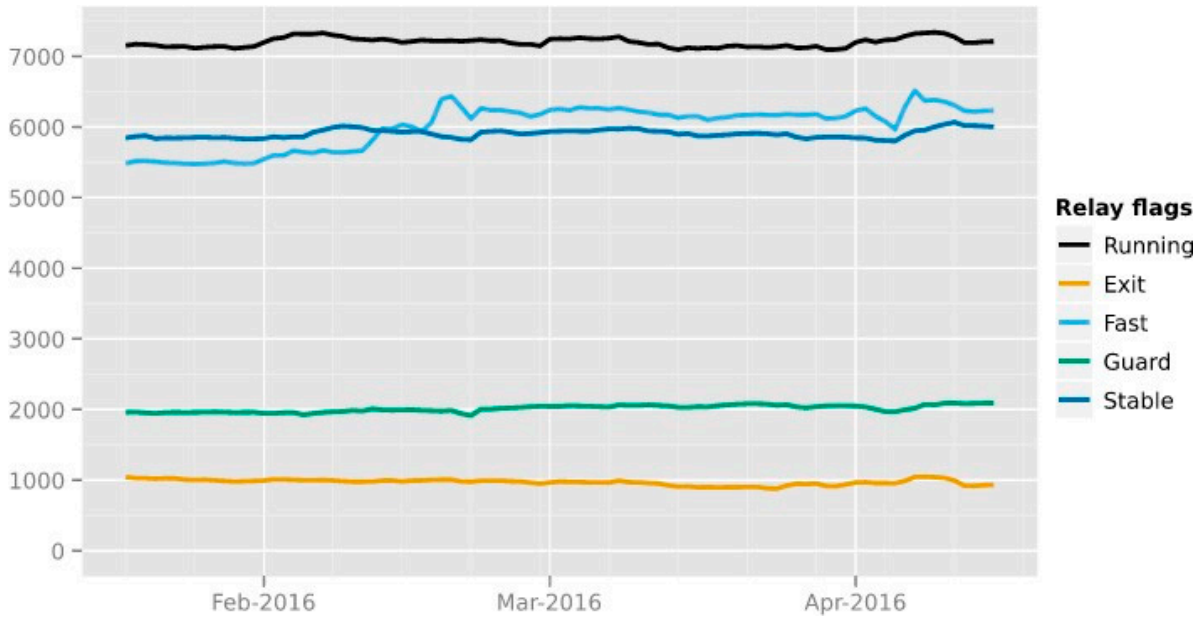


Figure 4. TOR router types and status evolution across time (TOR Project, 2016).Research on TOR:.

TOR is a project under active and continuous development process operated by a wide research community which is completely separate from the volunteers who are running the network. Nevertheless, the two community works closely in several cases such as development implementation, testing, security and performance analysis. Because of the popularity and wide use of the Tor network, several academic and industry people are getting more and more interested and carrying developments or performing research on and about TOR, frequently these researches topics include attacking passively or actively TOR platform or users. On the other hand on, works on anonymity enhancement, performance improvements or flexibility and scalability of the Tor network remain very limited.

During the last three years the research community has taken the challenge to improve the TOR security and performance more seriously especially after the so called “post-Snowden era” (Madden, 2014). Researchers proposed several enhancements which all contribute on the improvement of TOR (AlSabah & Goldberg, 2015). Those

proposals covered different areas such as congestion, scalability, routing, and security. Nevertheless, the most urgent challenges which TOR is facing could be divided into three major domains:

- As TOR is relying on an aging (componential breakable) encryption and integrity mechanisms for which global adversaries are or will soon be able and capable of compromising. Thus, the existing pretended cryptology guarantee such as onion construction (wrapping ciphers), keys exchange protocols and the integrity and authentication checks mechanisms should be reviewed.
- The delays caused by the congestion on the network and also by the heavyweight encryption and decryption.
- The identification of TOR user through internet and traffic analysis which leads usually to linking several intercepted communication to involved parties or linking multiple communications to a single user (Nia et al., 2014).

Several academic research works are currently carried out aiming to improving the TOR network. This formal approach was initiated by Backes et al. (2012) tackling for the first time in a formal way the security of TOR and proposing some solution for onion construction (wrapping) and key exchange protocols such as the 1-WAKE.

Regarding TOR circuits selection algorithm, research have tackled the vulnerabilities associated to the first generation of TOR circuit construction and selection algorithms, they ended by proposing (have been now implemented and applied into TOR) the concept of controlled selection Guard ORs and bandwidth/uptime caps (AlSabah & Goldberg, 2013b). Meanwhile, a particularly interesting research carried by Snader and Borisov (2012) in which they proposed two major enhancements; Firstly, they recommended after experimenting that bandwidth value which will be used for selecting the circuit should be measured centrally by the directory server using opportunistically sampling function rather than getting this value from ORs by allowing them to report their own capacity which could be misleading. Secondly, they propose a routing function to be integrated into the circuit selection algorithm which weights faster ORs more heavily but leave the user the choice between secure anonymity and performance.

Nia et al. (2014) proposed a novel anti-detection mechanism which used a function inside the crypto-system for data pattern generation and timely propagate the streams of data. Despite the effectiveness of the proposed system, it presents some difficulties in its implementation on the real world TOR and thus an eventual implementation could lead to radical changes on TOR which is risky. On the same year, Haraty and Zantout (2014) had presented TOR cryptosystem and detailed the operating principles and features. They had presented some threat that are facing the Second Generation Onion Routing without giving any solution or proposing an improvement. Also the ultimate aims were the resistance to traffic analysis, and better hiding the users' identities. However, since first and second generation TOR system relied on slow encryption mechanism and vulnerable routing and circuit establishment functions for which global adversaries possessing adequate resources was able to compromise and thus confidentiality, integrity and authenticity of data might be threaten.

Recently, a work carried out by Lee et al. (2015) tackled the existing TOR weaknesses especially those related to selecting the routing path for cells' shipping and the risk of fake intermediate and exit nodes and proposed a mitigation policy which was not implemented. AlSabah and Goldberg (2015) in their work entitled "Performance and Security Improvements for Tor" presented a state of current research directions focusing on the Tor network. They gathered in one paper almost all the TOR known design weaknesses and security flaws and improvement challenges facing TOR and enumerates the currently unresolved issues. However, the paper neglected two key points related to TOR: the multi-layers encryption challenges and the delays which is facing the real world TOR user.

On the mathematic and cryptographic side, only few research tackled the issue of proving the security of TOR's protocols and mechanism. A notable work carried out by Backes et al. (2012) entitled "Provably Secure and Practical Onion Routing" demonstrated the security and efficiency of TOR cryptosystem by elaborating formal framework for TOR. The work was considerably rich in content and divided into two parts: in the first part they introduced a cryptographic definition of global function behind TOR called the forward secrecy which was described as the optimal and perfect "in theory" to be implemented in the real-world TOR along with the One Way Key Exchange (1WAKE) Protocol used for the anonym authentication and Onion construction algorithm. The second part of their work was about proposing some improvements in the existing TOR integrity mechanism and also introducing new way of performing onion encryption into the CTR mode. However, this work neglected some primordial aspects related to TOR real world implementation flexibility and the overall efficiency. Backes et al. (2012) research remain to date the only formal and theoretical TOR analysis and the algorithms and functions introduced were

mathematically and cryptographically proved which will increase the TOR network reliability among the academic society and constitute a reference for further research.

2.4. Legal, Social, Ethical, Professional Issues and Academic Misconduct

During the last decade, researches interest on online security and privacy had sharply increased and a special interest was accorded to the TOR. The studies were conducted in different areas related to securing and improving TOR involving in some cases attacking it by designing attack that exploit the TOR security flaws. Currently, there is no agreement among the research community regarding the legal, ethical and professional norms. Unfortunately, several research activities have been considered as problematic not because of neglecting the ethical or legal aspects while performing testing or during data gathering (Soghoian, 2011). In this part of our work we will tackle these issues and debate some rules and limits which could resolve these problems.

2.4.1. Academic misconduct, legal and ethical issues:

This part highlights the current need of establishing and enforcing legal, academic and ethical norms for research works on TOR network by presenting two perfect examples which illustrate these issues and especially the academic misconduct of research studies performed on the TOR. The first work was published in 2008 where the second was in 2010 and went beyond the legal and academic regulation on gathering TOR real users' data putting at risk their security (Soghoian, 2011).

The first work entitled "Shining Light in Dark Places: Understanding the Tor Network" performed, in 2008, by McCoy et al. In order to gather TOR data, the researchers injected several TOR fake ORs working as exit ORs in TOR pretending to be volunteer. During several days, the researchers logged and stored the data extracted to study TOR traffic (McCoy et al, 2008). In their second part of the study, the researchers injected several "fake" entry ORs which allowed them to determine the IP addresses of a large number of TOR's users in order to determine TOR users number by countries (McCoy et al, 2008). Before initiating the work, the researchers, neither sought to investigate and evaluation the legality of their activities, nor, sought the guidance and approval of their academic authorities. Therefore, during publishing their work at the "Privacy Enhancing Technologies Symposium" they received hard criticism from both attendees and academic privacy community (Burstein, 2008).

The second example is the work entitled "Private Information Disclosure from Web Searches" which was carried out, in 2010, by Castelluccia et al. and aimed to reveal security flaws in internet browsers by sniffing cookies to reconstruct search queries (Soghoian, 2011). However, they didn't stopped in proving and demonstrating the flaws but used the acquired data to categorise and clustering TOR users and actively reconstructed the individuals' search history information for each user (Soghoian, 2011).

As a resolution for these kind of academic and ethical issues which could face the researcher in TOR network in the future. Dr Soghoian proposed in his paper entitled "Enforced Community Standards for Research on users of the TOR Anonymity Network" four rules to prevent researcher of falling into ethical and academic during their work (Soghoian, 2011):

- Research should be focused TOR and not TOR users,
- Minimize users' data collection and retention by ensuring that the data is examined directly and deleted after (seeking gather real data about should be secured),
- The researchers should not disclose any identifiable data regarding TOR users acquired accidentally during their work.
- Ensuring research's legality in the country where it is performed such as the legality of network monitoring for research purposes particularly in the occidental countries where online privacy and interception laws is exceedingly complex.

2.4.2. Professional and Social issues

Another type of excess related to the impact of research results on the use of TOR network (normal user) as some works can be exploited by malicious user to carry out hacking activities or communication between criminal members of organisation (narcotics, weapons and terrorism) resulting social and professional issues to the research works. An example of use of TOR to hide hacking activities and transmit stolen information was illustrated in the work intituled "Botnet over Tor: The Illusion of Hiding" (Casenove & Miraglia, 2014) in which the authors explain

how TOR is used by Botnet masters to provide C&CC (Command and Control Centre) anonymity, hence being undetectable and unsusceptible.

Chapter 3: Investigating current TOR Deployment

This chapter provides a summary of the TOR network deployment along with a critical investigation of the existing design and security weaknesses and also the attacks against TOR categorization.

3.1. Current TOR deployment

In this section, an introduction of TOR's main components and the current generation functioning principle, capabilities and limits will be provided along with a special focus on: onion encryption, circuits' construction and selection, security and overall performances.

3.1.1. TOR main components

TOR is a complex modular system composed from several sub-systems working together, Figure 5 illustrates the four basic components of TOR as follow:

a) *TOR Proxies TP (also called Client)*

Every TOR client runs a piece of software locally known as onion proxy (OP) which work as a link between the local machine and the TOR network and represent the client part of the anonymity system.

In

b) *Destination Server*

The TCP/UDP applications like a web service and is the second part into the anonymous communication with the client.

c) *Onion Routers (Nodes)*

ORs are dedicated software routers run by volunteers across the world which the work is to ship packets from client to destination. In TOR, the standard Transport Layer Security protocol (TLS) is used to ensure connection between ORs. The data is packed into equal-sized packets called "Cells"

d) *Certification and Directory servers*

The most important TOR components which hold almost all information about ORs, Circuits IDs, Network Status and Performances and other information which will be detailed later in this work.

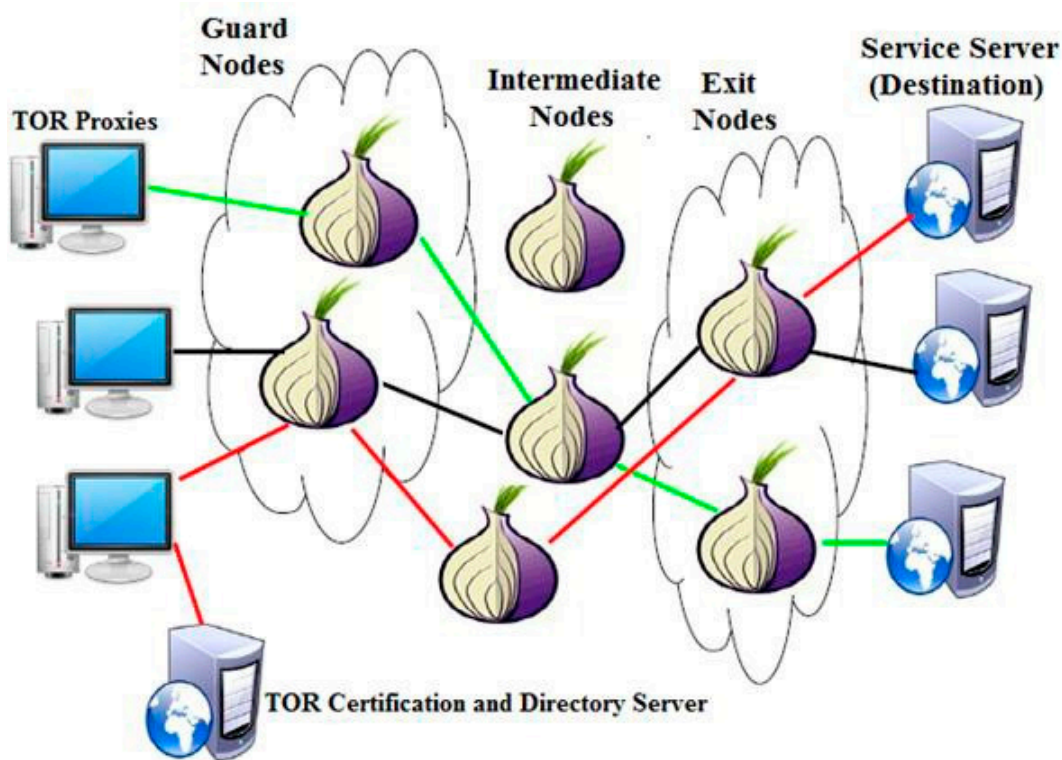


Figure 5. the TOR main components and overall architecture.

3.1.2. TOR communication mean

TOR uses fixed size cells of 512 bytes each to communicate internally and route the data from the client to the destination. TOR Cell is an abstraction of the normal internet transport protocols Datagrams format, it is constituted from two part:

a) Cell Header

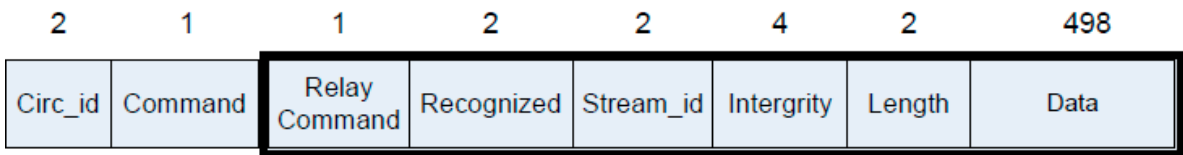
Have a size of 3 bytes, this part of the cell is never encrypted for transmission inside the TOR network in order to allow ORs in the circuit to access to the content.

b) Cell Data

Have a size of 509 bytes which constitute the transported data. This part of cell is always encrypted and only the next OR in the circuit have the decryption key and can decrypt it to gain access to the required original content or address. Nevertheless, the data part is, in reality, constituted from six parts as shown in the Figure 6. Further details about the commands and other fields will be detailed later in the next chapter.



a- TOR overall Cell format



b- Detailed TOR Cell format

Figure 6. The TOR Cell format and anatomy.

On the other hand, At the TCP/IP network level the TOR's Cells are embedded into TCP/IP packets. Therefore, having a fixed size cell will generate fixed size packets which is better for the camouflage of the exchanges. Alongside with the actual data, cells contain TOR system instructions for managing the network such as initiating new connection (circuits) or transmitting commands to other TOR parts. Figure 7 illustrates the use of TOR cell with the TCP/IP packets.

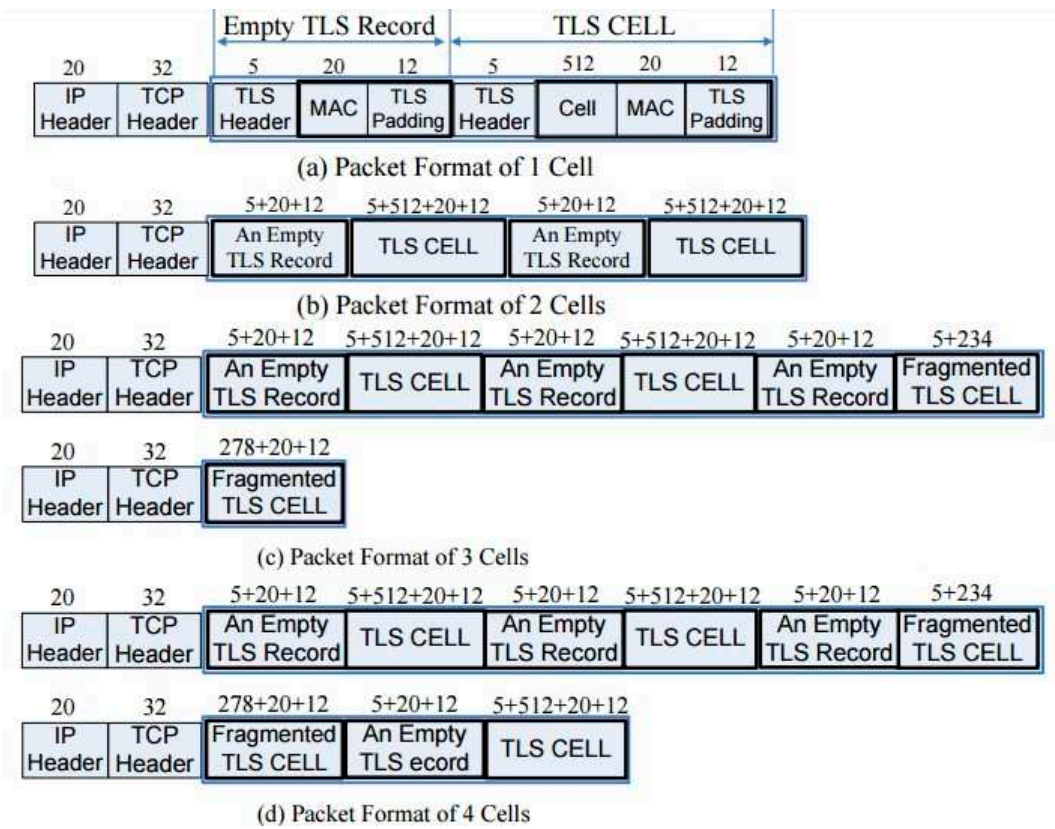


Figure 7. Packet Format at the Network Layer (Fu & Ling, 2009).

3.2. TOR working principle:

Considering the complexity TOR routing and crypto-system, investigating and understanding the current network design and implementation is essential before proceeding to proposing improvement and enhancement. This section aims to define the TOR structure and different protocols, mechanism or system used by TOR.

3.2.1. Acquiring, connecting and establishing circuit on TOR

After downloading a TOR browser which is done on the client side from the TOR project official website, a checksum is performed to check the downloaded application integrity and ensure that it has not been maliciously edited or tampered. Then, the client can then install TOR browser version depending on variant OS platform and thus establishing the OP (Onion Proxy). Later on, and after initialization the TOR client establish the first connection with TOR and acquire the TOR directories and needed data. This first connection is always done through the ORs called Bridges which are more than just simple ORs but they are set to accept new clients' first connections requests as they constitute a trusted ORs as they are secured and maintained by TOR team. After establishing the initial contact with one of the five management ORs, the OP performs the specific handshake steps required before connecting. Once the connection to the bridge (Guard OR) established, the OP will communicate the outcome to the one of the five directory node guaranteeing the services management again. The object of this connection is the construction of the circuit by selecting the involved three ORs and the order.

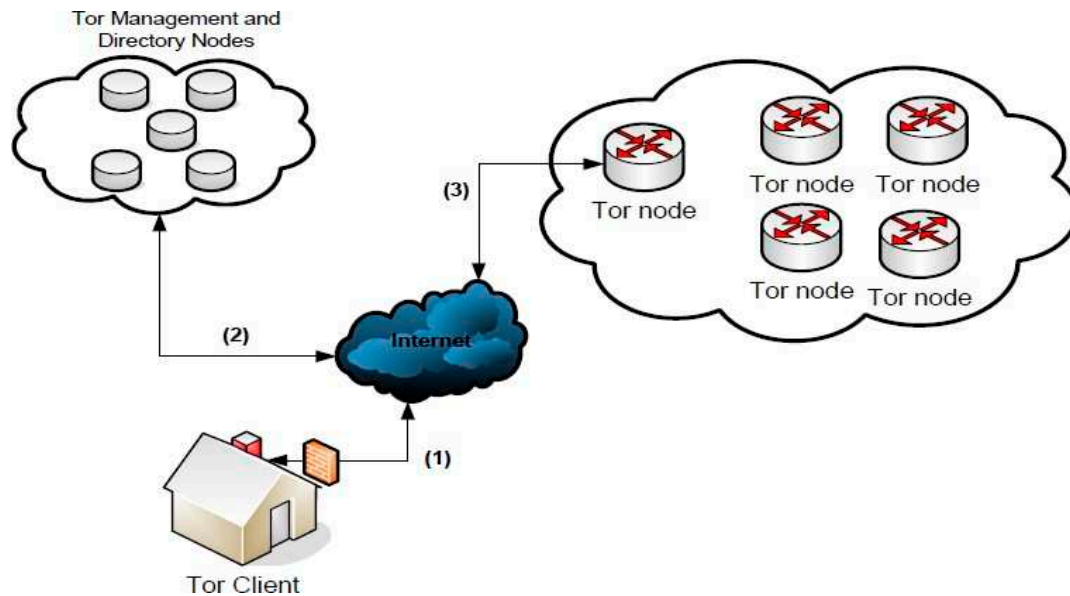


Figure 8. different steps for a user to get connected to a TOR.

3.2.2. Selecting and Creating a Circuit on TOR

To communicate with applications, the TOR cells need to be routed through a pre-selected ordered suit of ORs picked from a local cached directory (TOR Proxy) which is downloaded from TOR directory server. The suit of ORs is constituted from three ORs called circuit (or path) where the number of the OR is denoted as path length. The current use a circuit length of three. The circuit of three ORs is composed from: Entry-Guard Node (OR1), Intermediate Node (OR2) and Exit Node (OR3). The circuit selection should follow certain rules as the OR3 should possess the capabilities and information to execute the exit policy suggested for the TCP stream from the sender. Once the circuits selected by the client, it launches the procedure creating them over the selected circuit by performing one connection at a time. Figure 9 illustrates the TOR circuit creation and connection procedure.

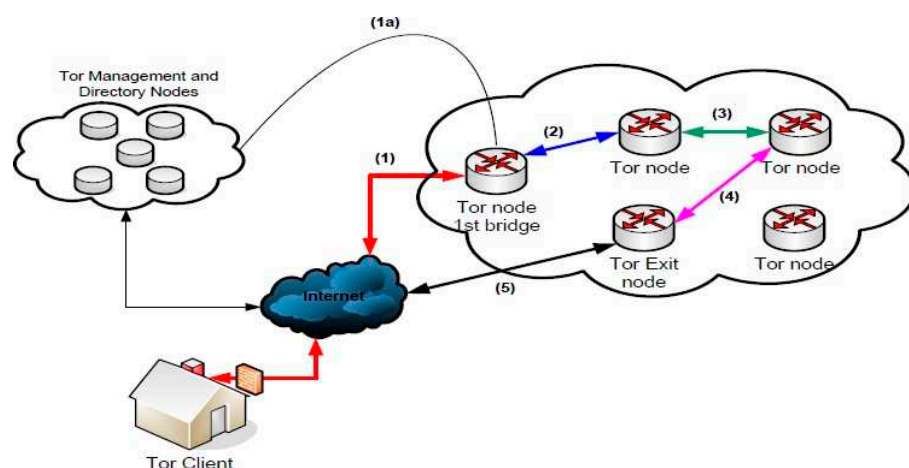


Figure 9. A Tor Circuit establishment and connection to a Tor Bridge.

3.3. Cryptography over TOR

The TOR conception allows each OR to relay data in form of Cells from senders to recipients or the inverses via the ORs which do not have a full access to the content (data) or even the entire path travelled or being travelled (routing circuit) inside the TOR or the initial source of cells captured before reaching their destination. This ultimate feature is achieved by the TOR cell combined encapsulation and multi-layered encryption mechanism called onion construction, during which each encrypted several time using the circuit ORs secret key order and sent throughout

the TOR network. Cell is then decrypted locally at the corresponding ORs which are the only able (following the pre-defined order) to reveal by decrypting (unwrapping) the identity of the next OR/ the final destination address after un- wrapping (decrypting) one layer of the cell (Fu & Ling, 2009).

To illustrate the TOR working principle, we consider the case of client downloading anonymously a file from a server. This example illustrates the complete process starting by establishing connection with a TOR bridge, communicating with the server via Tor passing through a circuit and reaching the destination for downloading the file as follow:

- The OP requires an Internet connection in order to able of establishing communication with the TOR network (DA),
- TOR Browser downloads the information known as consensus list which include the available bridges which will be used to initiating the routing circuit establishment.
- TOR establish the connection with the bridge (guard OR), using a specific handshake take place allowing the TOR directory server to authenticate itself to the client. In this phase a specific One-Way Key Exchange Authentication protocol named TAP (TOR authentication protocol) is used, this protocol is the spirit of the anonymity as it allow an authentication without asking for the client identity,
- TOR client broadcast a request of creating a circuit to all live (working) ORs on the list provided by the DA and select the ones to be part of the circuit in a predefined order following the selection algorithm.
- TOR client contacts the selected ORs which are part of the circuit in a secure manner to confirm the circuit and exchange keys.

After establishing the connection, initialising the browser and selecting the ORs constituting the circuit, the network information is updated on OP from the DA which transmit it on encrypted way so that the remaining ORs does ignore the ORs participating in the circuit. Thus, any OR only knows the directly linked two segments of the path; preceding OR which the cell come from, and the next OR which the cell will be forwarded to. This procedure is ensured as follows (Figure 10):

- The OP establishes a secure TLS link with the Entry (Guard) router called also bridge (OR1) using Create Cell C1 on an encrypted way using RSA and K1. The OP receive the confirmation from the OR1,
- To establish the remaining part of the circuit (two remaining ORs), the OP continues on the same method by passing though OR1. In fact, The OP establishes another secure TLS encrypted link with the second router (OR2) through OR1 using Create Cell C2 in encrypted way using RSA and K2. The OP receive the confirmation of link establishment from the OR2 via OR1.
- After a successful link establishment with OR2, the OP establishes the third link with the Exit router OR3 passing via OR1 and OR2 which establish the TLS encrypted link using RSA and K3. It is important to highlight that the links establishment procedure guarantee the forward secrecy (each OR is only knows the predecessor OR/client and the next OR/destination).

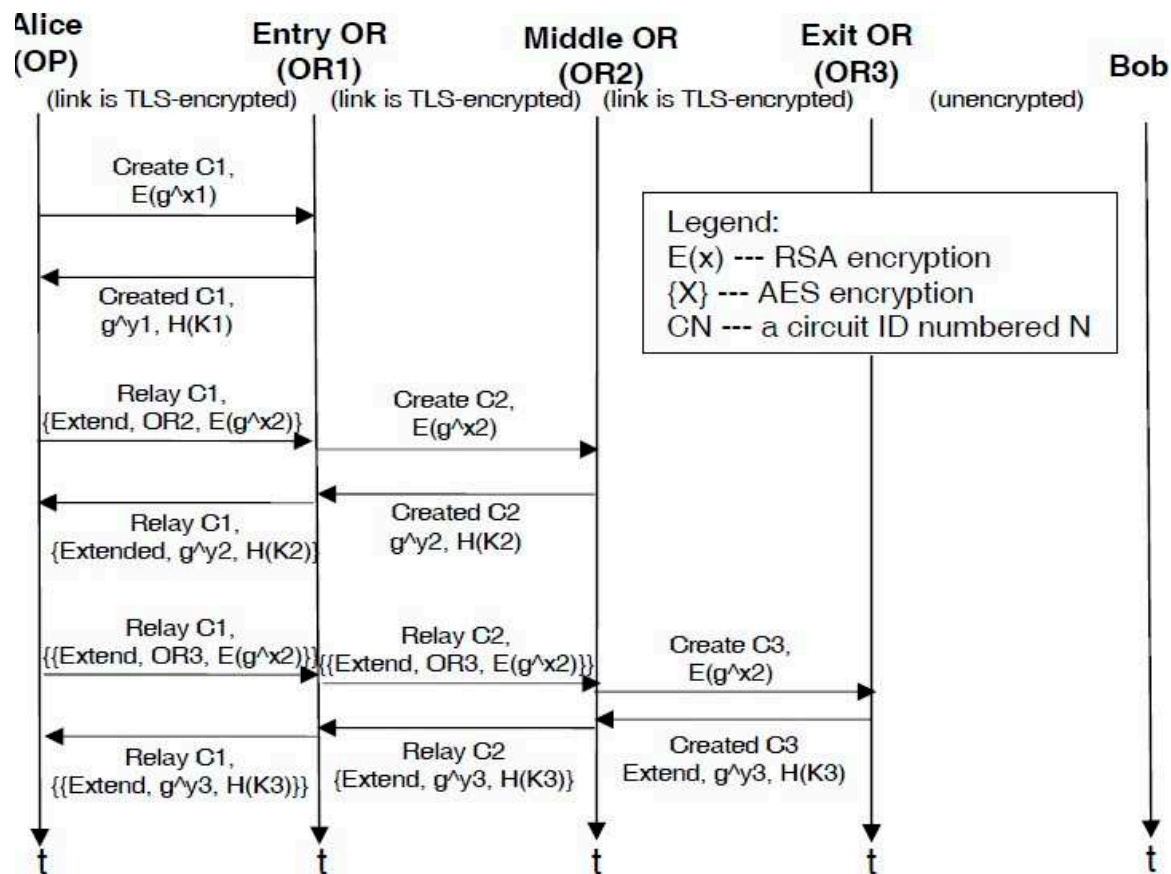


Figure 10. the TOR Circuit establishment and credentials exchange process (Boyd, 2011).

The above diagram illustrates in details the procedure of establishing a TOR circuit. In fact, TOR utilizes plenty of modern cryptography concept (algorithm, function and procedure), it first rely on public key cryptography to secure the data exchange when establishing the links between client and ORs, in this phase the information is transmitted in encrypted way using the receiver public key so the intended receiver is the only entity able of decrypting it (the only to know its private key). Moreover, TOR use heavily symmetric cryptography during both the circuit establishment and data transmission, it mainly uses the Advanced Encryption Standard (AES) algorithm. Other cryptographic concept such hashing and digital signature are also used to guarantee the data integrity, authenticity and non- repudiation on TOR (Fu &Ling, 2009).

3.4. Encapsulation and multi-layer encryption

TOR cryptosystem and encapsulation works as it is showed in the diagram. When TOR client needs to connect to TOR it start by establishing a secure connection tunnel with the Guard OR, it sends a Create_Cell to OR1 encrypted using RSA public key of OR1, so OR1 is the only able to decrypt and exploit it. The two remaining segment of the circuit will be built on the same way as the first segment.

Later after the circuit is built, the data being sent to ORs will be encrypted using a symmetric AES encryption which is more performance and efficient when it comes to big or sequential data transmission. TOR utilises AES with a key of 128 bits in CBC mode. In the latest version of TOR, the AES in a stream way with a key of 128 bits in counter mode (CTR) with an initialisation vector (IV) of 0 bytes.

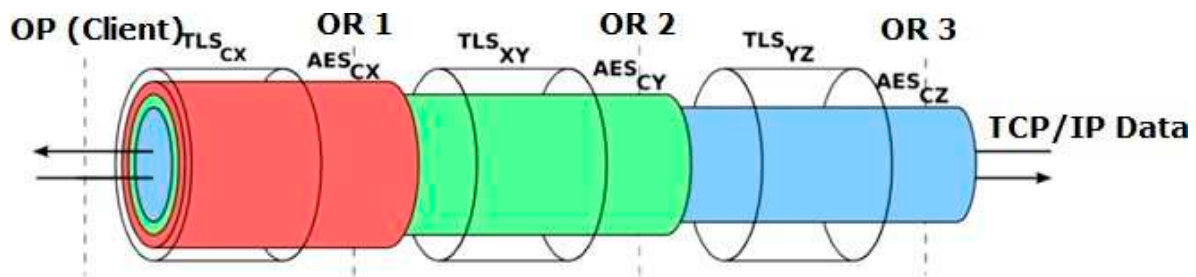


Figure 11. TOR data multi-layer encryption and encapsulation functioning.

When the Cells are being transmitted from the client to a destination through TOR, the packets, data is transported in form of Cells (509 bytes each) which are encrypted three times in an ordered way using the routing circuit ORs keys. The Cells are later routed through the ORs and each OR un-wrap (remove) one layer of the encryption to gain access to the address and information of the third OR. At the third (exit) OR, the last layer is decrypted and the OR direct the encapsulated packets to the final destination. This last communication could be either encrypted or clear depending on the used service by the initial client and supported by the destination server. TOR cryptosystem is not involved into securing this segment of communication.

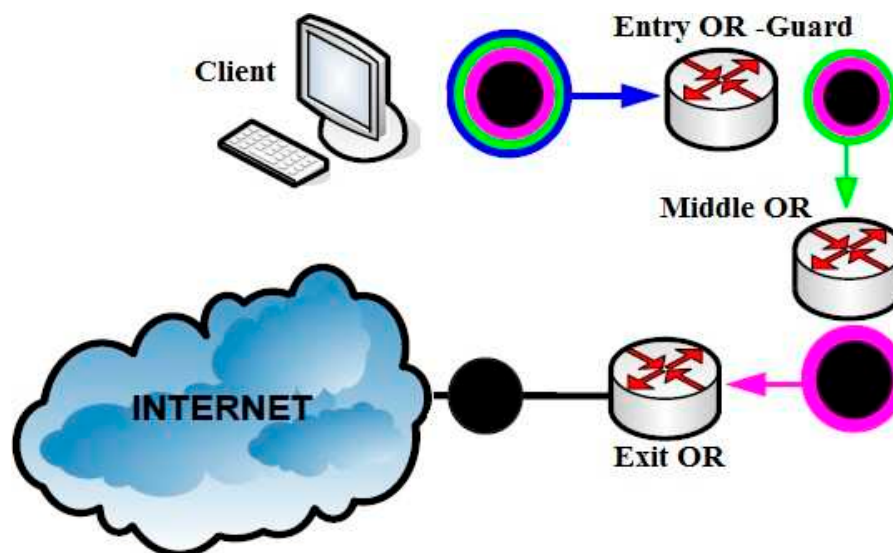


Figure 12. the un-wrapping mechanism across the TOR network.

3.5. Performances and Congestion Management

TOR is a big network of several thousands of routers. Therefore, to be able of controlling and managing traffic on TOR, ORs are equipped with a capabilities of logging, assessing and reporting performance and link status to the TOR directory servers. The mechanisms used the control and limit the amount of data that enter a portion of the network as follows:

- Real time link and traffic exchange: the ORs share in an real time basis the information about the network status and traffic proportion with the directories server, this information are used to determine the circuit and manage the congestion over TOR.
- Rate limit: ORs use token bucket configurable function that help to limit the bandwidth which an OR can use on the network.
- Circuit windows: TOR circuit relies on a flow control function which limit the quantity of data traveling through a circuit for a given duration. Both the OP and the Exit OR are responsible of maintaining that window which is usually initialised to the default value of 1000 Cells (500 KB). This is the maximum number of cells that OP and exit OR can send before an acknowledgment is received.

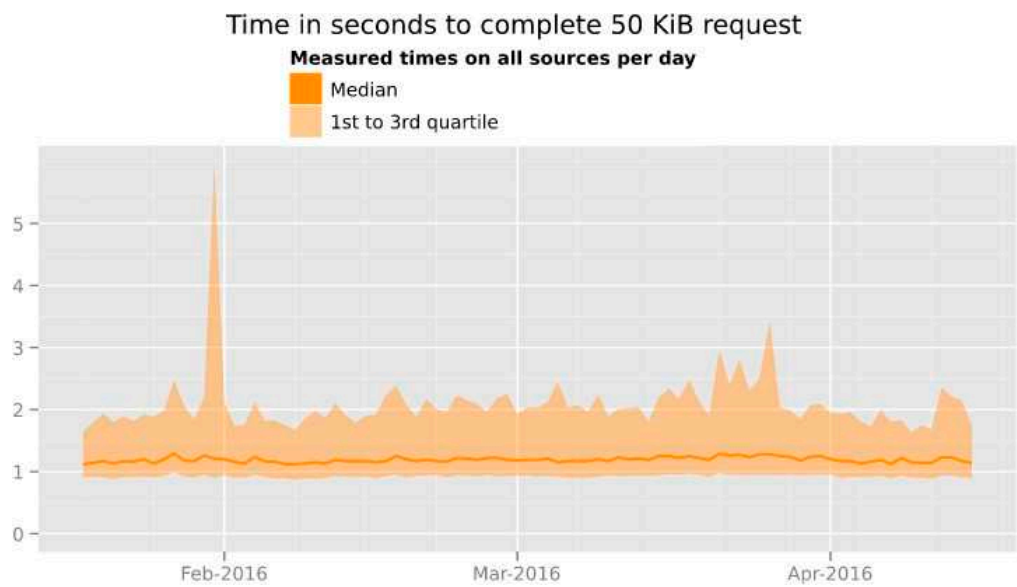


Figure 13. Time required to complete a download of 50KB on TOR (TOR Metrics, 2016).

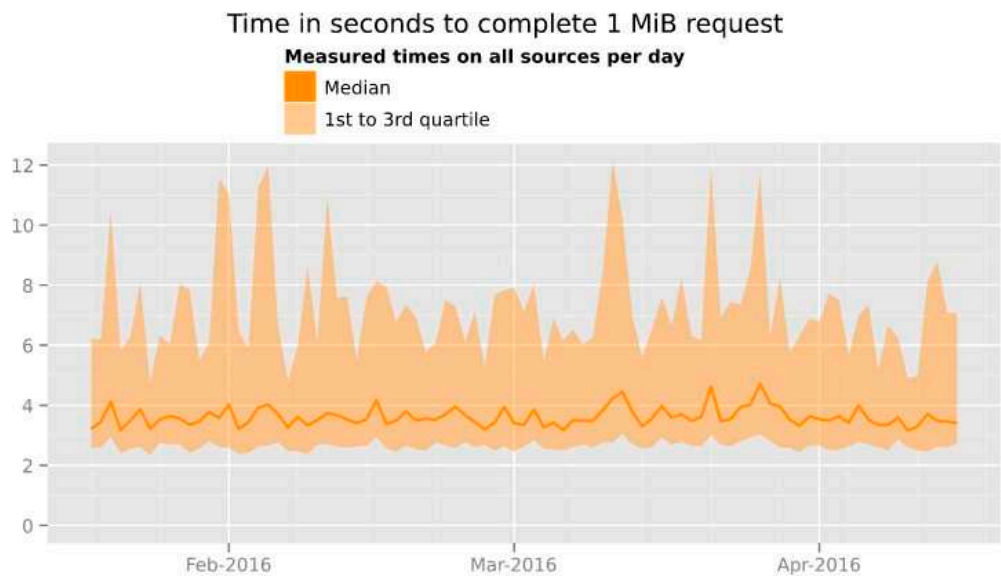


Figure 14. Time required to complete a download of 1MB on TOR (TOR Metrics, 2016).

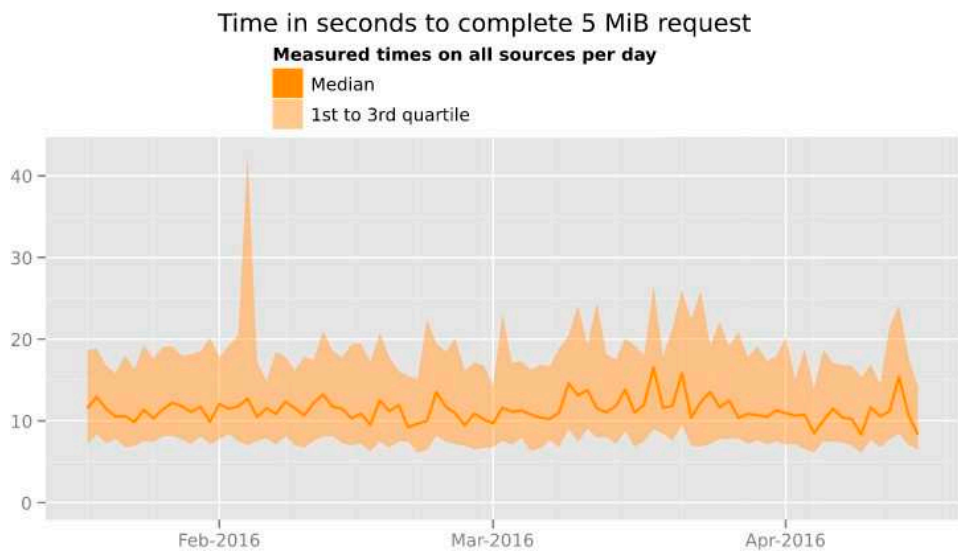


Figure 15. Time required to complete a download of 5 MB on TOR (TOR Metrics, 2016).

3.6. TOR Routing:

TOR traffic is routed in a manner to transit, before reaching the intended destination, through 3 ORs called Entry (Guard) OR, Middle (Relay) OR, and Exit OR. Nevertheless, some restriction are imposed to user such as the limitation of Entry OR number that a client can use which is normally set at 2 Entry OR. The Middle OR is chosen arbitrarily following a random function where the Exit OR also is picked randomly but from a selected list which is done following a pre-defined exit policy (specifies which IPs and ports). Circuit on TOR are automatically changed every 10 minutes but the client can change the circuit at any time. Moreover, TOR second generation introduced the multiplexing routing (Figure 16) which allow having several circuits over a single TLS connection.

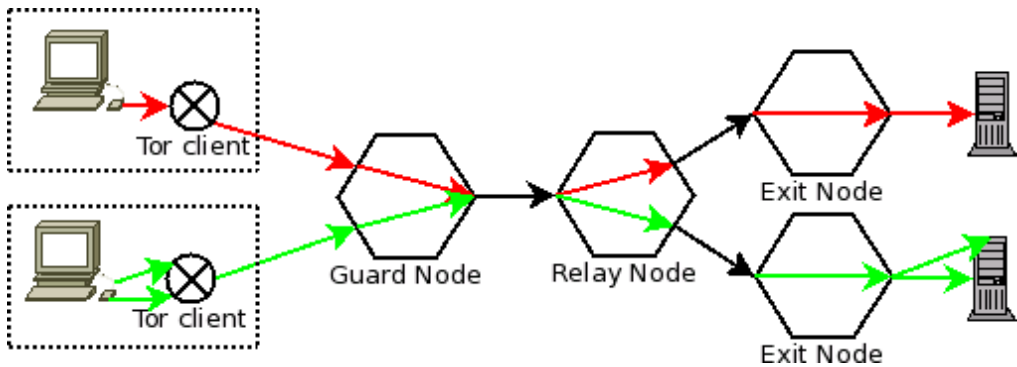


Figure 16. TOR multiplexing principle (Perry, 2007).

3.7. Circuit selection and scheduling:

TOR rely on a label-switching design that provide a multiplexing of many circuits over one OR. To guarantee that each TOR circuit is assigned the same amount (fair share) of the OR capacity, TOR used to employ a queuing mechanism called “round-robin” in which circuits are serviced in timing priority manner and thus all circuits get the same amount of the overall bandwidth. Although, McCoy et al. research in 2008 proved that in reality traffic distribution over TOR network was not uniform for all circuits as a small number of powerful and well positioned circuits were attracting a the majority of the traffic. To override this issue, Tang & Goldberg (2010) introduced a circuit scheduling prioritization mechanism using multi factors function (Figure 17) which is still implemented in nowadays TOR.

Network address	The IP address of the node
Node family	Administrator-configured equivalence class
Node bandwidth	Average, burst, and observed capacity; for most purposes, the average bandwidth is used, capped to 10 MB/s in order to limit bandwidth-based attacks
Node uptime	Time since node came online
Node status	Whether the node is <i>running</i> or is currently <i>hibernating</i> and so unwilling to accept connections
Exit policy	What, if any, types of exit node use is permitted
Publication timestamp	When this information was last received by a directory authority
Tor version	Version of Tor on the node

Figure 17. Circuit properties used in selecting paths.

3.8. TOR limits and vulnerabilities:

Besides the success of the first generation of TOR as online anonymity solution, there were some bad times for the project where a lot of weaknesses and vulnerabilities were revealed to be covered or just exploited my malicious people, researcher or even some law enforcement agencies to actively attack some TOR users. In September 2007, Dan Egerstad, a Swedish researcher, found out that it is possible to capture a lot of sensitive information such as

usernames and passwords for emails account though monitoring the data packets transmitted from TOR exit nodes (named later as Exit Node Attack), as at that time most of TCP/IP connections presented no encryption like SSL or TLS, the communications between the exit nodes and the final destinations were done in a clear manner (Schanck et al., 2015). Despite the fact that this attack did not compromised the TOR network overall security, it revealed in fact that the TOR is vulnerable to other type of attacks such as "Timing Attack", "Browser based attack", "traffic analysis attack" (Benmeziane et al, 2011).

As TOR gained in popularity, the list of enemies enlarged and the sophistication of attacked increased, where some attack aims to reveal TOR weaknesses to a later improvement. Others, mainly launched by TOR enemies are intending to compromise the network itself and de-anonymise users. There are obviously a several points of attacking TOR system. In this section, a summary of attack against TOR will be presented and discussing in extended detail. The reason of such focussing in the attack is because the proposed improvements aims to cover the vulnerability of TOR facing these attacks, and therefore, understanding the technology and the vulnerability used by each attack will help to address it properly. The attacks against TOR are categorised into the following categories:

3.8.1. Application Layer Attacks

These category of attack rely on feeding the TOR application with some data that causes bypassing the OP settings, expose/reveal user crucial information, or enable backdoors to exploit it. There are mainly three locations in which this attack can be launched (exit OR, middle OR, and final destination). A perfect example of this attack is the Exit OR spoofing and collecting data. This TOR vulnerability allows attacker to compromise non-https connections of TOR users. Another application layer attack related to the user itself is the Mozilla Firefox bundle attack (Zhang,

3.8.2. Intersection Attacks

Intersection attacks rely mainly on the successful correlation of several properties of a TOR user to achieve a matching between captured traffic and TOR user activity. This attack is particularly accurate when users use TOR in a slow, irregular (instable) connection speeds, in this case the remarkable delays and poor performance information will significantly help the attacker to narrow its intersection work which could potentially lead to de-anonymise the targeted TOR users' traffic (Schanck et al., 2015).

3.8.3. Traffic fingerprinting

AlSabah & Goldberg (2015) highlighted that it is possible to fingerprint TOR users. This type of attacks requires the control of the Entry OR on a TOR circuit. Since ORs part of the circuit are selected randomly, if an attacker establishes (connects) enough fake ORs into the Tor network, the chances are high that at least one of TOR users will choose a compromised OR as part of his circuit in the Entry level. During the establishment of a circuit, ORs have to exchange a big amount of data back and forth. In fact, Wacek et al. (2013) proved that by simply observing and looking for special timing patterns in cells transiting in each direction through the compromised Entry OR, then with a help of a machine-learning algorithms it is possible to determine with 99% accuracy whether the circuit is being used for ordinary Web-browsing, introduction point circuit (establishing first connection), or a rendezvous-point circuit (refresh). Moreover, this attack is particularly dangerous as breaking TOR's encryption wasn't even required (Johnson et al., 2010).

3.8.4. Circuit Failure and Selection Attacks

During the circuit construction, the Entry (guard) ORs role is crucial and this was exploited by attacker who force the entry OR to actively fail circuit by not extending to the following ORs. If instead the attacker manage to compromise an Exit OR and divert the traffic through any middle OR, side channel or timing attack could be launched to compromise the whole circuits having as Entry or Exit the compromised ORs. However, in case when the attacker continue to fail successive circuit extend for a determined duration the OR will be blacklisted by the DA and therefore underscored in term of reliability (Perry, 2007).

3.8.5. Timing Attacks

Time correlation is an advanced type of passive attacks which attempt by using information about connection time, and flow duration to correlate them to TOR users or connection via an Entry or Exit OR connection to an external internet address (Johnson et al., 2010). This type of attack can also be active when attackers inject their own timing patterns into the traffic. Several research studied and demonstrated that this attack can be extremely effective when the adversary possesses the adequate computing power and resources which remains crucial to bypass TOR's stream multiplexing strategy (Fu & Ling, 2009).

3.8.6. Path Selection Attacks

Several previous works have proved that it is possible to take advantage from the current TOR path selection algorithm flaws to develop efficient attacks. In fact, (Overlier & Syverson, 2006) proved mathematically and statistically that a client who repeatedly generates fresh paths can become more vulnerable as attackers gain a higher probability of controlling both the Guard and Exit ORs into the selected circuit. Moreover, to compromise the anonymity of TOR users an attacker can inject enough compromised (fake) ORs which increase the probability that a connection starts and ends within compromised ORs; this attack is known as Sybil attack (Dahal et al., 2015). Moreover, a much more efficient attack variant was proposed by Ling et al., (2011) which exploited the traffic load-balancing algorithm based on ORs which assess and report on a real-time basis the bandwidth and transmit to the TOR directory server (Perry, 2007). The researcher demonstrated that, by artificially changing the advertised bandwidth of ORs, an attacker is able to compromise almost half of TOR traffic while controlling only 10% of the TOR network ORs. This attack was carried out recently by US researchers working with the FBI to de-anonymise successfully several TOR users accessing illegal drug selling website "silk road 2" and since then known as FBI attack.

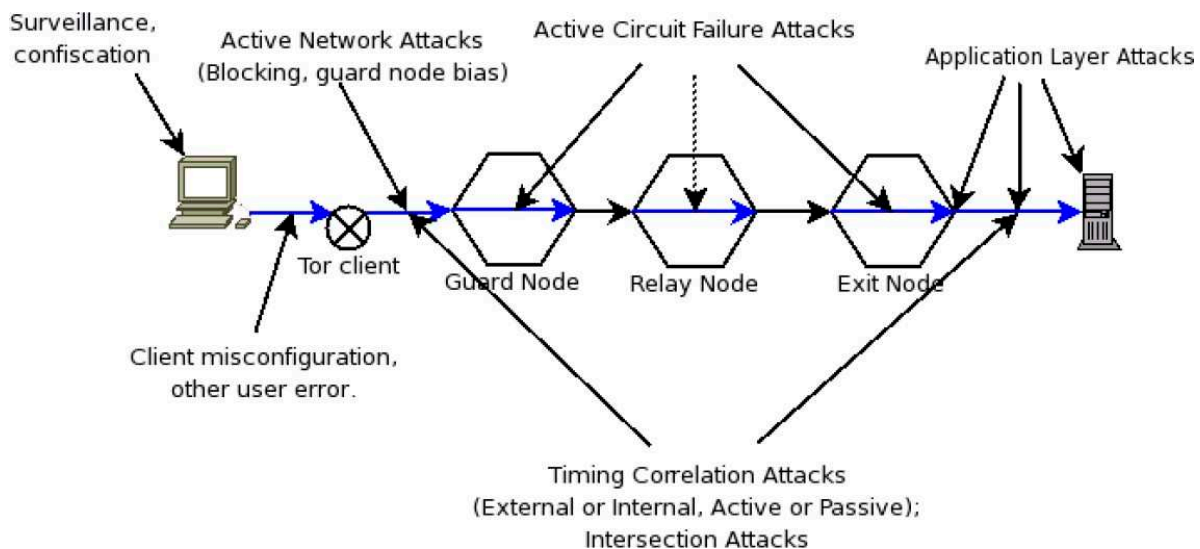


Figure 18. TOR points of attacks (Perry, 2007).

TOR still presents several vulnerabilities and potential points of attack; it is obvious that the system has no protection against a global adversary who can control both ends of the circuit, which was later in 2015 known as the FBI attack. A researcher had implemented in early 2015 an attack targeting initially the illegal drug market "silk road 2" in which they managed to introduce a number of fake ORs inside the TOR network, taking advantage of their status as academic and research organisations and the available digital assets and computational power. They implemented then an attack which diverted a part of TOR legitimate and real traffic through a compromised entry and exit OR. Once having data transiting via 2 controlled ORs out of 3, they launched a timing and traffic correlation attack to determine the last OR of the circuit. At the end, researchers were able to de-anonymise several thousand of TOR users, including a user suspected to be the manager of the drug market. This attack was mainly due to the unfair TOR circuit choosing and construction, which leaves the pseudo-random function the possibility of choosing a compromised Exit OR (Johnson et al., 2010). Moreover, the current TOR cryptosystem enables an adversary who possesses the computational power and technical deployment over the internet to observe and intervene actively on some fraction of network traffic by deleting, replaying, modifying or forging fake network traffic over TOR, which could compromise some fraction of the ORs and therefore compromise some directory servers. In fact, this flaw is mainly due to the

lack of node-to-node integrity check between TOR ORs which was initially sacrificed in favour of better multi-layer encryption performance which allow a more efficient TOR by reducing the delays caused by heavy-weight encryption.

Chapter 4: Proposed Improvements and Enhancements

4.1. *Related improving research:*

Previously, researchers had proposed plenty of improvement and enhancement techniques and methods. Where the majority were not realistic or unsuitable to be implemented on the real life situation, some on the other hand were implemented and produced the aimed security and performance goals. The perfect example of the academic research contribution into TOR is the work entitled "Anonymity and one-way authentication in key exchange protocols" (Goldberg et al., 2012) in which the TOR authentication protocol TAP was deeply reviewed to guarantee a Perfect forward secrecy. Furthermore, some others research outcomes were also implemented such as the separation of protocol cleaning from anonymity, the reviewing of mixing, padding in encryption algorithms, the use of multiple TCP streams in one circuit known as multiplexing, the local congestion control function and the enhanced end-to-end integrity checking. The following tree sketched by (AlSabah & Goldberg, 2015) illustrates and summarises the research activity on TOR.

4.2. *Problem tackled during this works*

This work look for improving TOR anonymity by enforcing some security mechanism and enhancing the performance of others. The proposed improvements will focus on improving TOR overall security and performances along with preserving and enforcing the current features which showed some weaknesses against the recent advanced attacks. This work state a clear statement in which an adversary external to TOR at any location with any capabilities should be unable to link (at large scale or locally) to link or identify any TOR user. This property is known as end-to-end unlinkability which is defined as guaranteeing the anonymity of the source regardless the destination's location. Note that the anonymity targeted is for both TOR sender anonymity and sender-receiver anonymity.

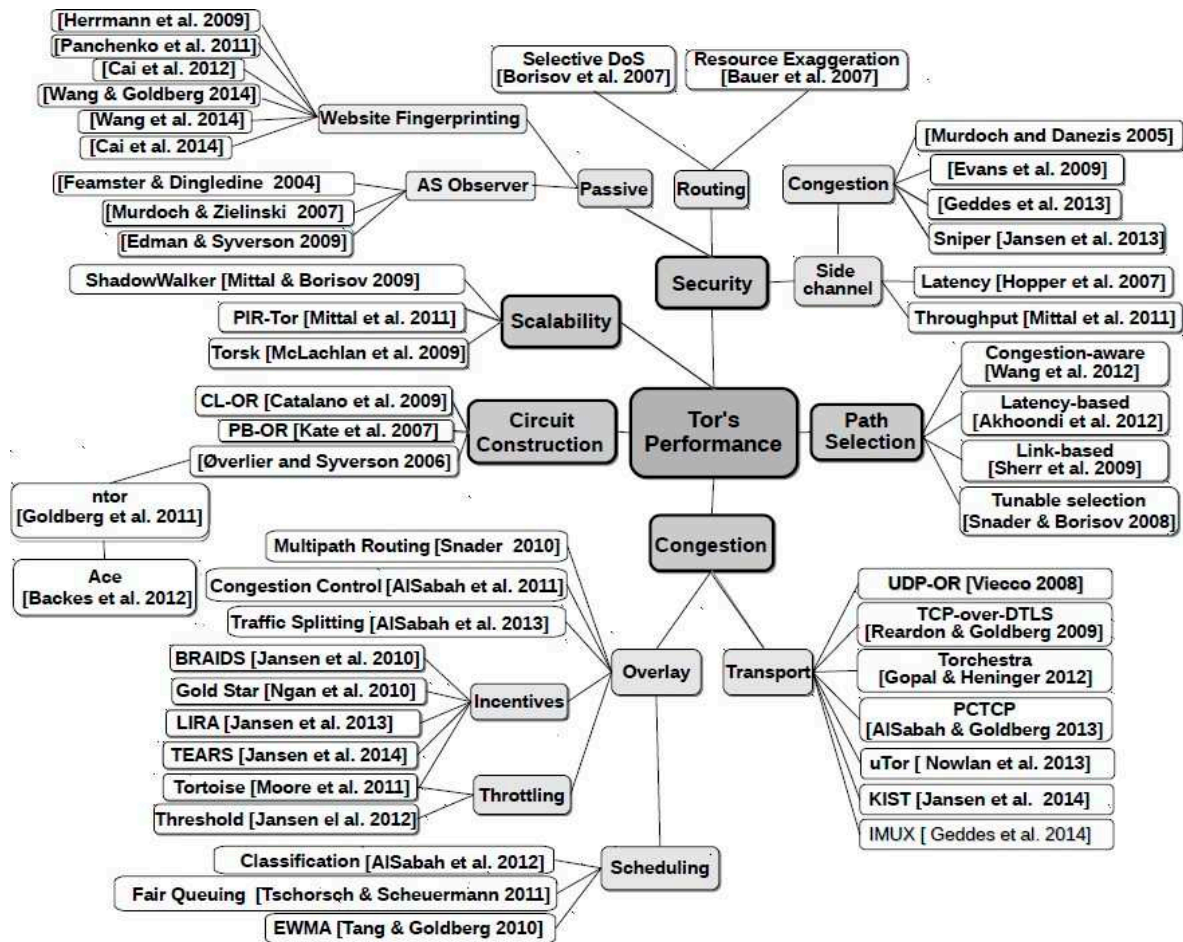


Figure 19. TOR research and improvements work tree (AlSabah et al., 2015).

Meanwhile, to preserve TOR reputation in the future, the security design should be reviewed and adapted to emerging threat and achieve better security and performances. In fact, several TOR flaws and vulnerabilities including the newly emerged ones remains untreated, the following is part of the desired properties and capabilities to be included in TOR:

a) *Cryptographic and performance enhancement*

The current TOR deployment, despite the fact that it is considered as a low-latency anonymity solution, is facing performance trouble which impact both the quality of the service and the platform usability and flexibility to be used in as much possible situation. Reducing delays is a priority and is closely related to the cryptography used over TOR.

b) *Circuit selection security*

An attacker, whatever his capabilities, should not be able to redirect any TOR client to choose a compromised circuit (controlled Guard and Exit ORs) or even modify the packet header to change path without being detected. The adversary should not learn forwarding information of uncompromised OR, OR's geographic positions, or the total number of hops on a circuit (length).

c) *Sessions' and Users' Un-linkability*

An adversary should not be able to perform timing attack to link Cells (packets) from different users or sessions, even between the same set of sources and destinations.

d) *Network Confusion and Diffusion*

An adversary eavesdropping on multiple links in the network should not be able to correlate two or more packets and determine that are from the same user by observing the bit patterns in the packet headers or data.

e) *Node-to-Node cells' authentication*

In addition to the existing Data **secrecy and end-to-end integrity**, inter ORs TLS authentication and Directory server authentication, TOR should include selective authentication check mechanism to identify Rogue Cells or any forged cells injected by attacker into the circuit for any intention (misusing TOR computing power and causing DoS, inject fake Cells).

4.3. *Proposed improvements and enhancements*

During this work, several improvements will be proposed covering different research directions which all aims to address the current design weaknesses, security and performances issues described in Chapter 3. The security is a wide term which in TOR context refer to three thing; information security (Confidentiality, Integrity and Authenticity), Anonymity and un-linkability. The information security part is tackled in this work by proposing an improved AES mode guaranteeing the Confidentiality and the Authenticity simultaneously, where the anonymity and un-linkability are tackled by reviewing the TOR wrapping mechanism, circuit construction and routing within TOR which appear to be not relevant for the security but in reality it's the most important factors for anonymity and un-link-ability. Moreover, this work tackle the performance issue in TOR and the related security flaws and vulnerabilities, as several attacks against TOR exploited the delays and the poor performances (timing attack, users' clustering, and path selection attack).

4.3.1. Multi-layer encryption improvement

In this part, we assume that the AES (Advanced Encryption Standard) functioning principle is well known. Nevertheless, we will describe it in details at the appendix. The purpose of this part is to investigate the suitability of replacing the current AES implementation in CBC mode used for cells' encryption by the authenticated-encryption OCB mode (Bogdanov et al, 2014).

OCB (offset code-book) is a new and revolutionary implementation of AES block cipher guaranteeing authentication along with the traditional confidentiality (privacy) of the user data, this type of ciphering id called *authenticated-encryption scheme*. Moreover, OCB mode is surprisingly and remarkably fast as it achieves authenticated encryption quicker and consuming almost the same duration as oAES encryption only CTR mode. Therefore, by adopting OCB the user can achieve in the cheapest way two out of three information security goals in an optimised and secure manner as OCB is considered as a simple cipher and resources efficient cipher when it comes to implementation in either hardware or software (Krovetz & Rogaway, 2014). In nowadays cryptography OCB solve three majors cipher issues:

- OCB eliminate the problem of authenticated-encryption with associated-data (AEAD),
- The OCB nonce required to encrypt and decrypt should not be necessary random as it utilise a counter,
- OCB can encrypt data of any size without padding it to any convenient-length and therefore save some precious computing power.

TOR like all others Cryptosystem present cryptographic vulnerabilities related to its corporants, and one of these is the unauthenticated Cells traveling throughout the network leaving the possibility of forging fake Cells an inject them inside the network for malicious purposes. The ORs authentication mechanism remains insufficient. To introduce a node-to-node authenticity check on TOR, two options are available; the first option is an AES mode providing both privacy and authenticity separately (CBC, CTR and others) which perform separately encrypting and then computing the associate authentication using two different keys, the cost of having authenticated-encryption is thus the cumulated cost of encrypting with the cost of MAC.

a) *Working principle of the OCB mode*

AES-OCB is a block-cipher with a block length and a key (K) of 128 bits each. It also uses a nonce (N) of 96 bits and an associated incremented counter value (Δ). The OCB detailed working principle is as follow (algorithm in Appendix 2):

- First the plaintext M is divided into blocks of 128 bits each $M = M_1 \dots M_m$, Here there is two cases; the data size in bit is a multiple of 128, or there is a remainder and therefore the algorithm require a padding.
- Secondly, a Checksum of 128 bits is calculated $\text{Checksum} = M_1 \oplus \dots \oplus M_m$ and will be used later during

the authentication process.

- Thirdly, an initialisation function "Init" take place and using the nonce N which is concatenated with a 32 bits constant value to produce a 128 bits value called "Top". Later, the $K_{top} = E_K(\text{Top})$ is computed and Stretched to produce the 256 bits value $\text{Stretch} = K_{top} || (K_{top} \oplus (K_{top} \ll 8))$ (left shift by 8 positions K_{top} and replace the empty by zeros). The value $\text{Init}(N)$ which is the initial value for Δ .
- Fourthly, for each block "i" the increment is called to increment the Δ , XOR with the M_i , and encrypted using the key K and the algorithm AES-OCB as showed in the scheme. Later, the output of the encryption in stage 4 is XOR again with the Δ to produce C_i . The authenticated ciphertext is $CT = C_1 C_2 \dots C_m T$.

Afterward, the authentication value which is 128 bits length is computed by processing the associated data A which is XOR with the value of Δ for each block in the same way as the encryption part and later encrypted using the same key K . the result of all the blocks is XOR together to produce a 128 bits length authentication value Auth . Finally the checksum of the initial data is XOR again with the Δ , encrypted using the key K and then XOR with the authentication value Auth to produce a final authentication Tag " T " for the whole data (Krovetz & Rogaway, 2014).

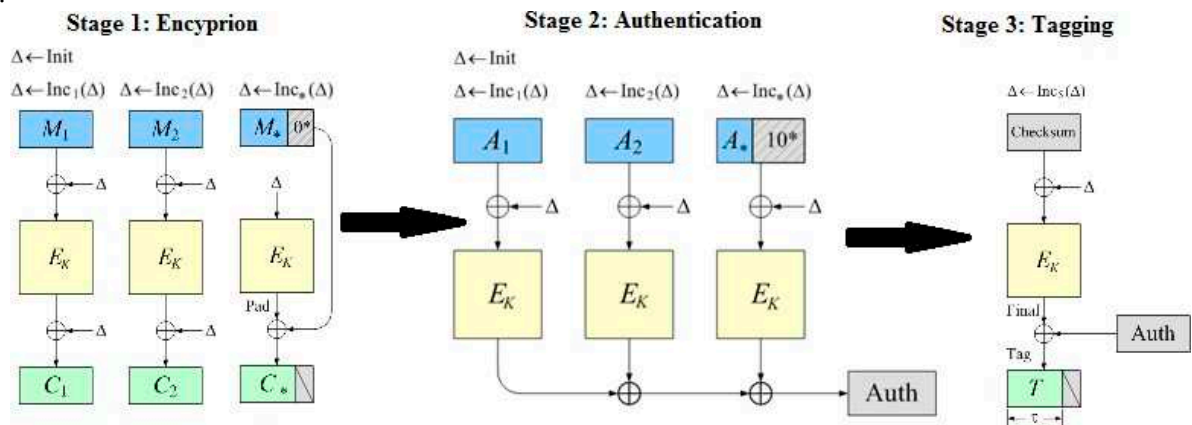


Figure 20. OCB mode functioning schemes (Bogdanov et al, 2014).

The Decryption under OCB mode is faster and simpler. By having given K , N , and CT , the receiver recover the initial message M following the normal decrypting way. Then, the authentication tag T is re-computed and compared with the received one to determine the authenticity of the received message (Krovetz & Rogaway, 2014).

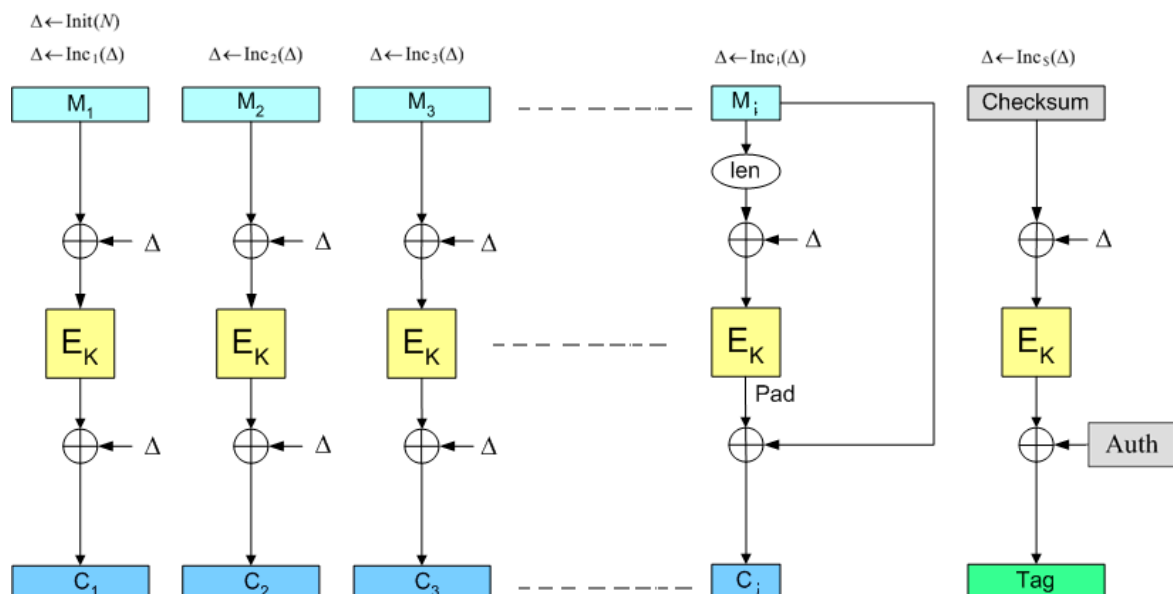


Figure 21. AES-OCB version 3 encryption and authentication operations scheme (Krovetz & Rogaway, 2014).

There is three different implementation (versions) of OCB mode. The adopted version for this work is version 3 which is the most optimised version in term of operation number and computational power. Although, the performance improvement will be relatively small, it remain crucial for TOR to adopt an authenticated-encryption for onion construction. In fact, in addition to the performance enhancement, implementing AES in OCB mode will bring the following properties:

- Fully parallelizable operations of the block ciphering can be performed simultaneously. Thus, OCB is very efficient and suitable for hardware encrypting at high network speeds.
- Block-ciphering scheme make it strong and resist better to the new timing attacks which the other mode like CBC would be vulnerable.
- OCB is a single key scheme as it use the same key for encryption and authentication which make it more efficient in term of memory use.
- OCB can process any data size without requiring it to be a multiple of the block length. Moreover, no external padding function is used and thus it economise time as there is no bits- waste in the ciphertext due to padding.
- The main computational function used beyond the block-ciphering is XOR which is very time and power efficient function (three 128 bits XOR per block).
- OCB can be perfectly used into memory-limited systems as the main memory cost the amount needed to hold the AES sub-keys.

b) *Choice justification*

An authenticated-encryption scheme enable two parties sharing a secret symmetric key to communicate in a manner that ensures both privacy and authenticity. AES implementation in OCB mode is designed to be time (time consumed to perform encryption) and resources (processor and memory) efficient in both software and hardware. In fact, the algorithm is perfectly adapted to restricted environments requiring accuracy and pseudo-synchronization along with providing provable security and authenticity (Krovetz & Rogaway, 2014). During this work we will start by assessing OCB Security and performance versus others competitor integrated authenticated-encryption modes. The use of an incremented nonce for each encryption and the decryption process by OCB is one of the major strength. In fact, It is required that the nonce should be unique (not necessary random, secret or unpredictable) for each message but OCB rely on a counter value which ensure that each nonce is different. Thus, the importance of the unicity of the nonce is crucial to maintaining perfect authenticity and privacy.

On the other hand, OCB competitor scheme and particularly CCM and GCM which offer an integrated authentication along with encryption will be assessed in similar testing environments during this work. Moreover, the traditional approach for achieving authenticated-encryption which rely on composite functions (encryption following by MAC or MAC followed by encryption) will be assessed alongside with OCB competitors evaluation, two implementations of both CBC and CTR mode followed by MAC computing will be performed to serve as a reference on the performance evaluation. In this work the implementation of CBC and CTR mode will not use separate keys, in fact we will use the same 128 bits key for encrypting the plaintext and then to calculate the associated MAC of the resulting ciphertext. Nevertheless, for security measure the CBC IV (Initialisation Vector) will not be derived from the key but instead it will be generated using a different function. The following table, summarize the different mode of implementation and the picked candidates for potential adoption instead of the existing CBC/CTR and are: OCB CCM, GCM, and EAX:

scheme	date	high-level description	standard
✓ EtM CTR	2000	Encrypt-then-MAC (and other) generic comp. schemes	
RPC	2000	Insert counters and sentinels in blocks, then ECB	
IAPM	2001	Seminal integrated scheme. Also IACBC	
✓ CBC	2001	Concurrent with Jutla's work. Also XECB	
OCB1	2001	Optimized design similar to IAPM	
TAE	2002	Recasts OCB1 using a tweakable blockcipher	
✓ CCM	2002	CTR encryption + CBC MAC	NIST 800-38C
CWC	2004	CTR encryption + $GF(2^{127}-1)$ -based CW MAC	
✓ GCM	2004	CTR encryption + $GF(2^{128})$ MAC	NIST 800-38D
✓ EAX	2004	CTR encryption + CMAC, a cleaned-up CCM	
OCB2	2004	OCB1 with AD and alleged speed improvements	
CCFB	2005	Similar to RPC but with chaining	
CHM	2006	Beyond-birthday-bound security	
SIV	2006	Deterministic/misuse-resistant AE	RFC 5297
CIP	2008	Beyond-birthday-bound security	
HBS	2009	Deterministic AE. Single key	
BTM	2009	Deterministic AE. Single key, no blockcipher inverse	
✓ OCB3	2010	Refines the prior versions of OCB	ISO 19772

Figure 22. AES implementation modes comparison (Bogdanov et al, 2014).

c) *Cryptographic features comparison against competitors*

To evaluate the features of each candidate in addition to the practical performances, this work rely on the following points to determine the suitability of the authenticated encryption mode, the features are summarised in the following table and divided into three major part:

Table 1. Authenticated encryption features comparison (Krovetz & Rogaway, 2014).

Feature	CCM	GCM	OCB
Security Proved	Yes	Yes	Yes
Online ability	No	Yes	Yes
Key requirement	128 bits block size	128 or 64 bits block size	128 or 64 bits block size

- **Provably secure:** all the three modes are proved to be mathematically secure by assuming that the used with block cipher (AES) is pseudorandom permutation. As far as the cryptography permit, AES is proved secure and thus both three modes of implementation are absolutely secure.
- **Online message processing:** this feature is crucial for the suitability of the mode as the modes should be able to process data without knowing the whole length in advance as the TOR have no pre-set or pre-defined data length. Moreover, this feature is highly desired for a memory restricted environment which is the case of ORs in this part, CCM mode fail to achieve the set baselines.
- **Cipher requirements:** CCM mode is developed to only work with ciphers using block size of 128 bits, while GCM and OCB can work with cipher using different block size (64/128 bits).

Nevertheless, this feature will not affect the CCM mode as the block size in TOR is 128 bits which is anyway more efficient and better for performances.

4.3.2. The Encapsulation approach (Onion wrapping method)

TOR use Cells as mean of transporting TCP/IP data throughout the network to the exit OR which will be in charge of transmitting it to the destination under TCP/IP protocol. Currently, TOR perform a multi-layer encryption-encapsulation which mean that the initial data is placed into fixed size cells of 512 bytes each (509 bytes for data and 3 bytes for header) and then encrypted three times using the three ORs constituting the routing circuit keys into the

inverse order. In other words, the whole data along with the next OR address or the final destination is encrypted three times (figure).

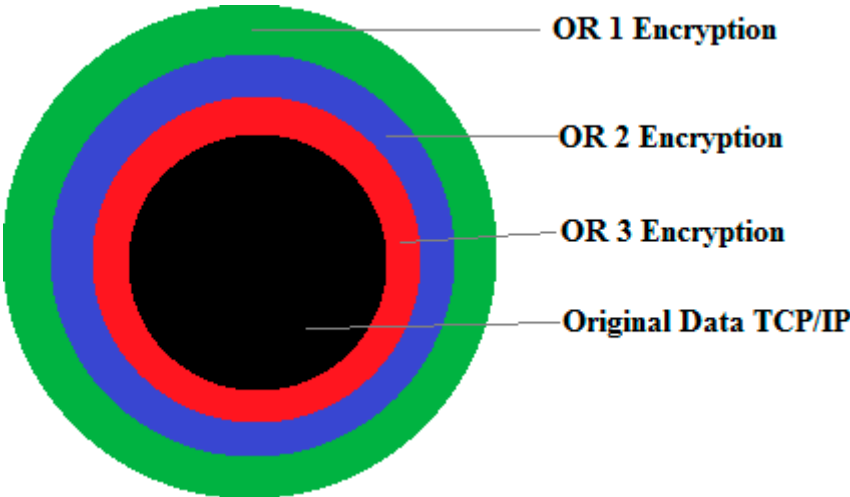


Figure 23. the onion multi-layers encryption approach.

This approach of multi-layer encryption is the hearth of the TOR system as it allow only the ORs part of the Circuit and In Ordered way to have access to the information related to the next OR in the circuit or the final destination of data and thus achieving the anonymity of the sender (figure). However, giving the delays caused by heavyweight encryption of relatively big data this mechanism of encapsulation and wrapping became problematic as it, in one hand slowdown the performances and in the other hand was proved mathematically that this approach does not bring additional security to the system. In fact, in cryptography encrypting the same data using the same function (algorithm) several time using different keys will give the same security of encrypting it once using a composite key which is the aggregation of all the keys (not the addition but it is mathematically determined).

The current TOR Onion Construction Algorithm

Get the data and the final destination
Construct the circuit and get keys ready
Divide data into TOR cell size
First layer:
Data= Original Data+ Final Destination Address
Data= Encrypt (K-OR3,Data);
Second Layer:
Middle-Add= Encrypt (K-OR2, Data + OR3 Address);
Third Layer:
Data= Encrypt (K-OR1, Data + OR2 Address);
Data= Data+ OR1 Address
End Onion Construction

To summarize, improving the encapsulation and wrapping on TOR will not only improve security (privacy and anonymity) but also enhance the network overall security and resilience as the current relays on TOR are being exploited by several attacks (correlation and timing attacks).

d) **Proposed Improvement:**

Instead of multi-encrypting TOR Cells several time to produce an onion wrapping which only circuit ORs will be able to unwrap, we proposed a much efficient and time saving approach which perform a full encryption of the

whole original data including both Exit OR address (Cell Header) and TCP/IP (Header and Data) in the first phase using the Exit node AES shared-secret Key. Then, for the remaining ORs (Entry OR and Middles ORs) only the cell header will be encrypted. In cryptography, encrypting several time data using different keys (k_1, k_2, \dots, k_n) is equivalent to ONE encryption using a composite key K . Thus, the current TOR encryption of the whole Cells several times is useless and cause performance slowing down only as one layer of strong encryption n is enough.

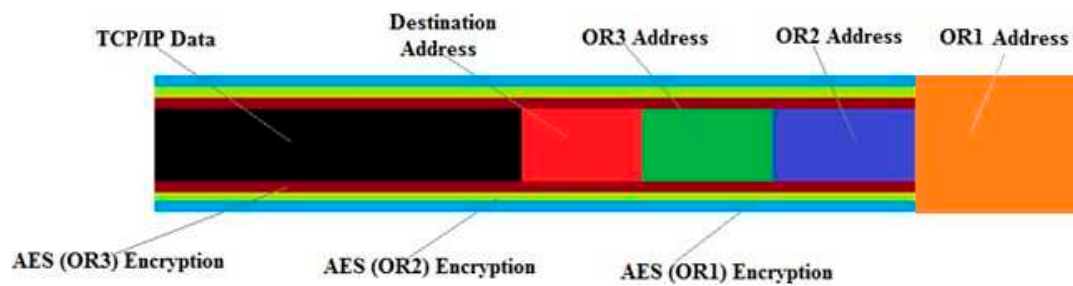


Figure 24. TOR current cell multi-layers and encapsulation approach.

Moreover, the current TOR cells structure is vulnerable and should be reviewed, we propose that only the internal Cell Header contain the Circuit ID, where the external ones (OR1 and OR2) should contain only the address of the next OR and Command. Giving the fact that TOR is managed locally, including this kind of information cause redundancy causing the slowdown of the operation and also leave the circuit ID exposed to threats especially when a fully compromised OR is a part of the circuit.

The proposed approach of encapsulation and onion construction work as follow:

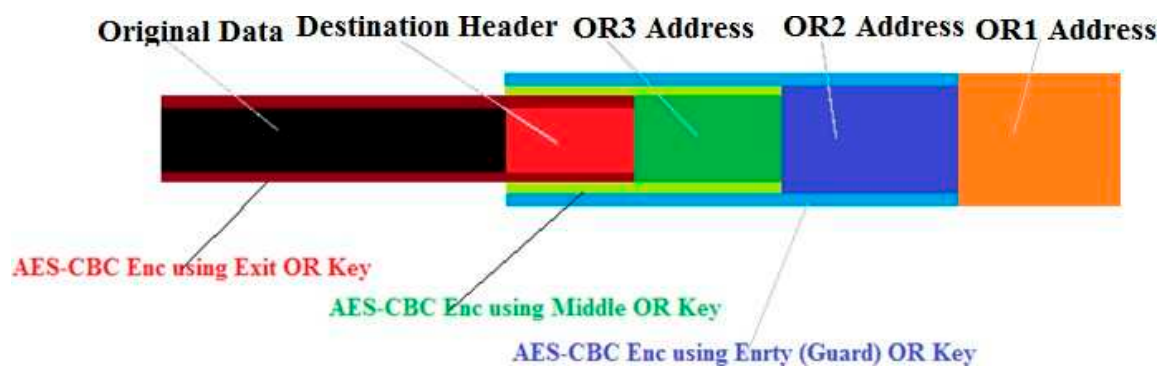


Figure 25. TOR proposed cell multi-layers and encapsulation approach.

The Proposed TOR Onion Construction Algorithm

Get the data and the final destination;

Construct the circuit and get keys ready;

Split data into TOR cell size;

First layer:

Data= Original Data+ Final Destination Address

Data= Encrypt (K-OR3, Original Data);

Exit-data= Encrypt (K-OR3, Final Destination Address);

Second Layer:

Middle-data= Encrypt (K-OR2, Exit-data + OR3 Address);

Third Layer:

Entry-Data= Encrypt (K-OR1, Middle-data + OR2 Address);

Finalizing Cell construction:

Data= Data+ Entry-data+ OR1 Address;

Basically, the proposed wrapping algorithm will economise (save) a crucial time and componential power. In theory, two (02) round of encrypting data of 509 bytes will be saved along with preserving the same security of the existing TOR. In cryptography encrypting the same text several layers using different (but same size) keys will have the same security of encrypting only one time using the composite key. As AES is provably secure and the key used is also assumed secure enough to suffice on one layer of encryption.

4.4.3. The number of intermediate ORs

TOR users anonymous online activities is mainly due to two computing technology Cryptography and Routing, TOR utilizes a series of ORs and makes users' data traveling through a number of hops before it reaches the final destination. By ensuring that each OR have not more information than its predecessor and its successor in the circuit, TOR hide the origin or the destination of the cells containing data and therefore guarantee users' anonymity. Given the aforementioned principle, it is obvious that the more is the number of ORs into a circuit, the better is the source of data (client) is hidden and thus anonym, tracking back the communications will become very complex and the majority of times just impossible to perform. Meanwhile, the number of ORs influence the connection performance as the long circuits cause more delays (latency) and running interactive applications requiring time precision connection becomes impossible. Hence TOR developers, were seeking the best trade-off between a secure connections that enables perfect anonymity while keeping connections latency bearable. Following several research and testing, TOR developers finally adopted the three (03) ORs circuits length which the current TOR deployment use. When a client is communicating with a server, the data is routed through three intermediate ORs before leaving the TOR network and reach its destination. This choice is defined as the optimal balance between security and usability of TOR.

A continuous debate has been raised regarding the appropriate circuit length especially after the 2015 FBI attack which with the help of researchers were able to control, at several occasion, both the Guard and the Exit ORs and therefore performed an advanced attack to de-anonymise several TOR users including "Drug website Silk-Road-2" owner. As consequence, current TOR short three (03) ORs circuit length will be critically reviewed in this work and several improvement strategies will be considered including increasing the length of the circuit and adopting new routing strategies such as controlled exit OR which will be discussed later in this work. TOR developers' intention behind the choice of a default three ORs circuit is to provide the best balance between the security and performance. In fact, this choice include an Entry OR, an Exit OR and an additional OR aiming to obfuscating the link between the entry and the exit ORs in such way that even if an attacker is able to compromise either of these ORs, the middle

OR will constitute the last layer of defence as the attacker can only observe encrypted traffic and it cannot directly deduce the identity of the user. Nevertheless, with the rapid increase and development of the computational power and cryptanalysis attacks, this defence layer become meaningless and can be compromised by performing timing correlation attack.

a) Proposed Improvement

A systematic thought is that increasing the circuit length further would produce an increase into the TOR security. Unfortunately, this operation will incur a significant impact on TOR performance and penalise further the network as the more ORs are involved in transporting one cell, the more times the same cells are relayed in the network before reaching its destination. To determine the impact of increasing circuit length onto TOR anonymity, we will experience different cases in which the variables will be either the number of OR into the circuit or the routing methodology itself such as controlled exit OR selection and network link-status depending selection.

In this work, we implemented a dynamic TOR circuit construction function which have as input the natural number P which is the length of the circuit, then we uses this function to measure the impact of a longer or shorter circuit on the performances. In the proposed function we proposed different circuits building approach for evaluation purposes, also rebuilding function was modified along with the initial function following two main criteria: time interval and circuit performance. Furthermore, as TOR connections terminating in the public internet, the weakest points for attack in the circuit is obviously the Exit ORs (last OR in the circuit). We introduced the notion of "Controlled Exit OR" in the circuit construction in which the algorithm responsible of defining the ORs which will take part in the circuit is adapted to choose the exit ORs from a pre-defined list which reflect in the real TOR the list of trusted ORs. This proposed solution is expected to reduce considerably the FBI attack success against TOR despite the number of rogue (fake) OR inserted into TOR network (Steven et al., 2011).

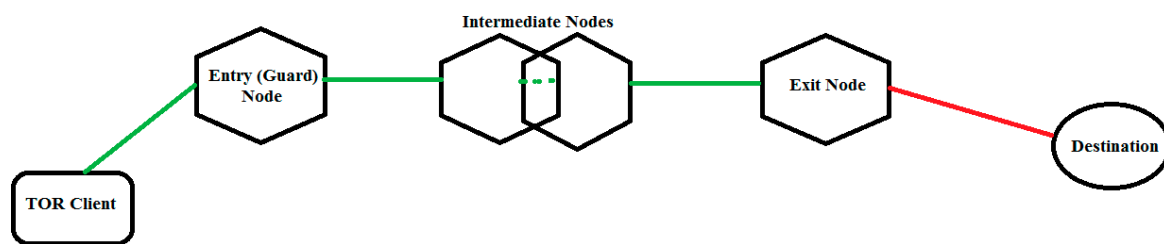


Figure 26. Dynamic TOR circuit length scheme.

4.4.4. Circuit ORs' Selection Approach

In this work we will improve the path selection process by proposing a novel path selection hybrid-algorithm relying on varying path length, controlled exit OR and real-time performance assessment functions. We also employ ORs parameters from a simulation of TOR to compare the proposed algorithm efficiency against.

In the current TOR implementation, ORs for circuit construction are selected at a uniformly random basis aiming to guarantee that ORs are selected uniformly and thus increases the probability and the uncertainty of an attacker trying to de-anonymise users by guessing the ORs used in a particular circuit. Although, because of the heterogeneity in resources caused by ORs of different capacity (computing power, bandwidth) and the emergence of a new classes of attacks, the selection algorithm was edited by TOR development team in the second generation of the network, the following changes were introduced in form of exception in the algorithm:

- No OR should be used in the same circuit more than once,
- ORs in the same circuit should belong to different class of TOR network,
- A special treatment for co-administered ORs is introduced by marking them as the same family,
- Directory Authorities will assign flags to ORs basing on the following parameters: performances, status, position and role.

Moreover, some important features were added following the attacks on TOR circuit selection in 2014. In fact, ORs selection algorithm was again changed in such way that entry OR (guard) is only selected from a subset of ORs classified as "entry guards" and which are particularly "trusted" and continuously authenticated to check the status. The entry guard subset is a group of ORs which are constantly active, having a bandwidth of at least 250 KB/s (Dingledine et al. 2014). In reality, the implemented selection algorithm allowed 3 ORs guard to be assigned for a user for a period of 30 to 60 days and used in combination for all circuits. However, due to the limit probability of

attacks and the impact on the TOR overall traffic homogeneity, this algorithm was abandoned to allow client the use of only one entry OR for the same period of time (Dingledine & Mathewson, 2015).

On the other hand, current TOR circuit selection algorithm states that selecting the remaining ORs on the circuit (Middle and Exit) is proportional to the available bandwidth. This choice aims to ensure that powerful ORs are chosen more often. Nevertheless, the bandwidth information which TOR base on for making decision is being advertised by the ORs themselves which leave the opportunity for rogues ORs to advertise false data in order to acquire more traffic. Logically thinking, if a capable attacker with significant resources is able to inject an important number of rogue ORs having the best performance, it will be able to re-direct a significant amount of TOR traffic via these ORs (which will be middle ORs or exit OR of the selected circuits) and therefore being able to perform a time-analysis attack to determine the User as the Entry OR information will be already disclosed for the Middle OR (Steven et al., 2011).

a) *Proposed Improvement*

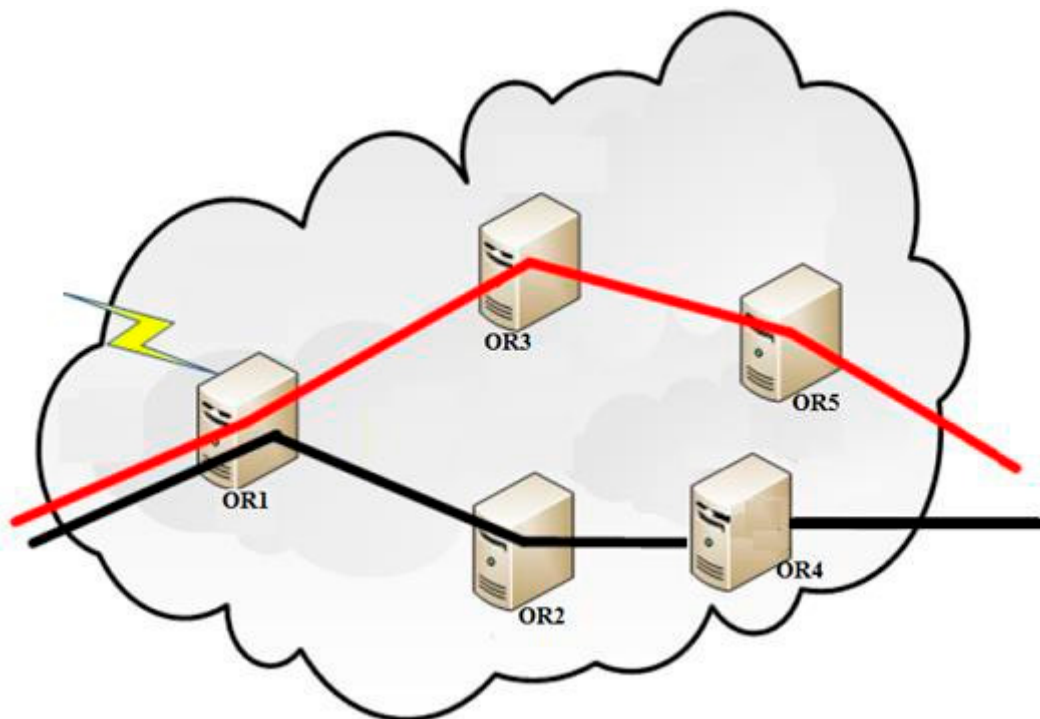


Figure 27. TOR circuit selection scheme.

The proposed improvement for the circuit selection algorithm is twofold:

First, we introduce a new sub-function in the section algorithm imposing the selection of the Exit OR from a pre-defined subset only. This subset will be completely different from the entry guard subset (no common ORs) and contain more ORs. This method is called during this work "Controlled Exit" which will be implemented into the emulation platform for test purpose. It is evident that such method will have an impact on traffic homogeneity, but the security enhancement will be greater than the overall performance. Moreover, the proposed method will help into protecting the TOR hidden- services from being disclosed during the attacks.

The Modification introduced to the current Circuit ORs Selection Algorithm

Proposition 1: Controlled Exit OR:

- Defined a list of Exit ORs (5 ORs out of the 20 ORs constituting the simulation network,
- When choosing an Exit OR the DA should assign an exit ORs from the pre-defined list,
- An Exit OR can be a Middle ORs,
- The Entry (Guard) ORs cannot be Exit OR.

Proposition 2: Best Possible Circuit (B/W)

- This algorithm bases on the real time data provided by the DA regarding the links status assuming that there is other traffic and not all the link have the same capacity.
- We use Dijkstra's algorithm to determine the three ORs constituting the shortest path by relying on the bandwidth as a determinate factor.

The second proposed improvement is related to the choice of the ORs basing on the advertised capacity (bandwidth). The misleading information that rogue (fake) ORs can provide during the selection process could be fatal for user security. Thus, the directory authority responsible of the collection of such information and processing should not trust this information and rather evaluate or estimate itself the capacity of each router and therefore faster ORs will not have any more a higher probability. In this work, the concept of prudential-processing is introduced in which the calculation of ORs' capacities is done in a real-time basis relying on both provided information by the ORs, historical status and credibility. This method will be implemented and tested into the emulation platform.

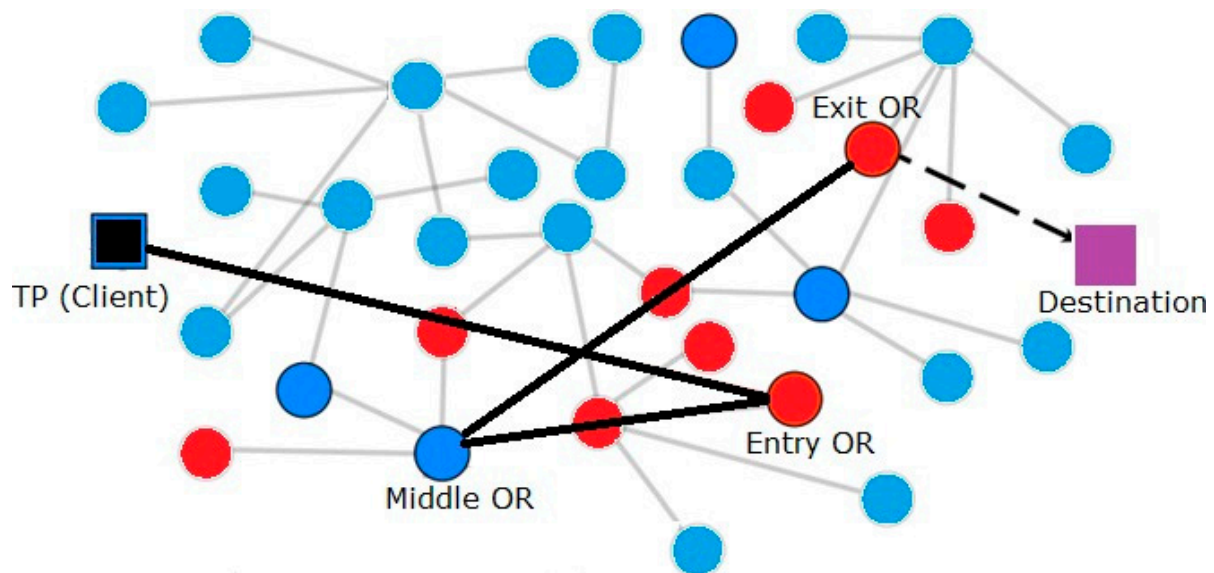


Figure 28. the proposed controlled exit approach for TOR circuit selection.

b) Dynamic circuit construction with traffic management

To tackle the delays and security issues related to the choice of circuit ORs, we introduced an enhanced circuit selection algorithm named Controlled exit with congestion-aware ORs Selection at Client side. In this work we tried to adopt and implement a circuit construction method proposed by (Wang et al.,2012) in which the multi-criteria circuit selection algorithm rely on both the status and performance data and real time indicator. First, TOR's default bandwidth-weighted OR selection algorithm is used to construct circuits. Then, the proposed algorithm will use an opportunistic and active-probing function to calculate the circuit's latency value.

4.4.5. Cells' Multi-circuit routing (limited to three Circuits)

Multipath routing approach on TOR has been previously tackled by Snader et al. (2010) in context of improving TOR networking performances. The research simulated a case of downloading data fragment of 1MB each over a dedicated TOR simulation network. The files were divided into blocks of 512 Bytes and routed from the sources to destination over multiple circuits. The proposed mechanism was working as follow. An algorithm assign each Client (OP) two different Entry and Middle OR, the mechanism used the same Exit OR for both Circuit (Figure 29).

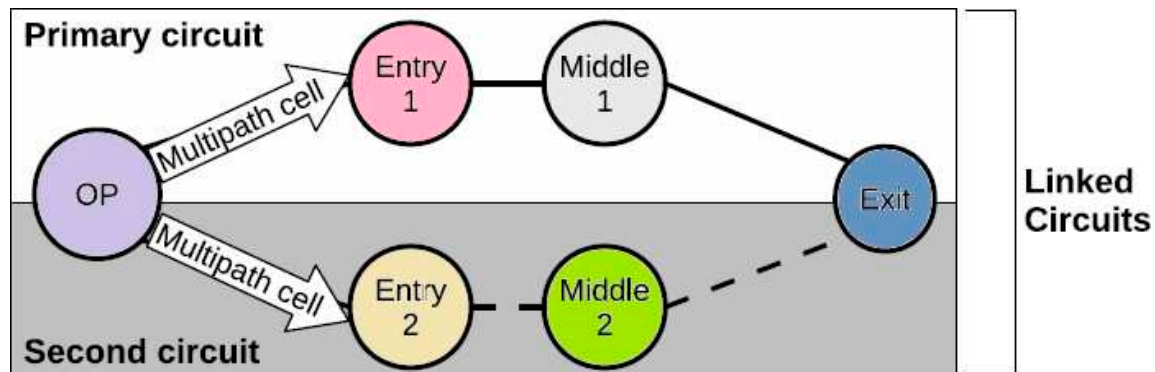


Figure 29. TOR existing multi-circuit routing mechanism.

This research observed that the security and throughput of the routed traffic were significantly enhanced. However, two circuits performance was less well than single circuit as the chances of choosing a slow OR as part of the circuits double and thus the median transfer time increased for the two circuits. Nevertheless, the research highlighted the fact that the security enhancement of such proposition is also considerable and also that the risk of including a compromised OR will be certainly be affected if this function is used.

a) *Proposed Improvement*

The current TOR deployment each Client (OP) is assigned by default three different Entry (guard) OR which will be the only Entry guard that this client can have (use) for a certain duration. This research will rely on this feature and implement an algorithm which will generate for each linked Traffic (having the same destination server) three different circuits. This Multipath routing has introduced to enhance security for TOR users and also the Entry and Exit OR themselves. TOR client starts by building a three different circuits using the following algorithm:

Multi-Path circuit construction Algorithm

For each TOR new Connection do:

// from the assigned 3 Entry OR for

1- Select a different Entry (Guard) OR from the list of OP three possible Entry ORs;

// a random sub-function will be used

2- Select Randomly a Middle OR;

// from the assigned 3 Entry OR for

3- Select a different Exit OR from the list of Exit OR;

// complete the circuits construction

4- Perform hand-shake, authentication and keys exchanges

// for each data being sent to the same destination

5- Divide the Data equally to 3 parts and send each consecutive 3 cells throughout the 3 different circuits.

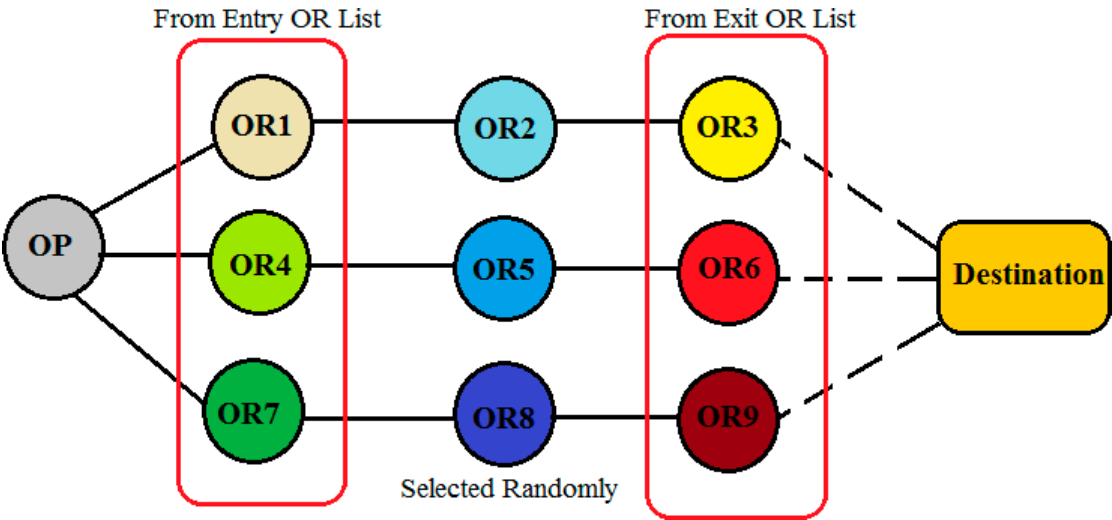


Figure 30. the proposed TOR multi-circuit routing mechanism.

Chapter 5: Implementation, Testing and Results

Testing is a crucial phase of any system development or improvement and require a particular intention as the results of this phase will determine if the system should or not goes to life or the improvement are valid and thus could be implemented in the real world system. The proposed improvements and enhancements should be measured and evaluated only by adopting a scientific approach for comparison which can determine whether or not the obtained results are satisfying and can bring the intended contribution along with preserving the system main features and resources. On the other hand, Cryptographically Improving TOR design decisions, such as adopting new ciphers, integrity checks mechanisms, routing method and encapsulation technique, is a complex process which require a reliable test-bed.

An obvious testing approach is to use the real TOR network to conduct the different testing and measurements as TOR is open-source software, hence easy for researchers to implement and run experiments. Nevertheless, this option is ethically not suitable, as during the testing and experimentation other TOR user’s. Therefore, to test the efficiency of the proposed improvements and enhancement for a potential incorporation into TOR (cryptosystem or

routing system), it is essential to simulate the TOR on a virtual environment along with preserving the same behaviour and accuracy of the real TOR network. To achieve this, we adopted an existing TOR simulation platform and introduced several modification and code changes which reflect the proposed improvement.

This choice is justified by the fact that testing on the real network will incur serious privacy concerns to TOR users' security and adopting a non-standard test-bed will negatively impacting the confidence on the obtained results. Therefore, after having carefully evaluated the scientific risk and the feasibility along with the required componential power, testing efficiency and accurate which the test-bed should fulfil. This work adopt a two phase testing approach starting by evaluating the efficiency of the candidates alone (with just program implementation and testing into JAVA compiler) and then implement the selected (accepted) improvement into the simulation TOR along with the native function to produce a comparison and eventually validating the results obtained.

5.1. Simulation test-beds design and adoption

To practically demonstrate the performance enhancement brought by the proposed improvements, two isolated test-bed platform were designed/adopted. The first emulation platform is adopted to run both current and improved TOR simulation as the first simulation will run a native TOR with the current deployment, parameters and configurations in a simulation network of twenty (20) ORs (Nodes) running virtually along with AD (directories server), three OP (clients) and Destination Server. Among the ORs, three are considered as Guard (entry) ORs. The testing will starts by assessing and recording the performances of the current TOR version which will serve as a reference to evaluate the efficiency of the proposed solutions, later several modifications into OP code, DA and ORs will occurs to reflect the proposed solution.

On the other hand, and before implementing the improvement into the TOR simulation platform, a cryptographic implementation and testing of the candidate cipher modes to replace the current CBC mode is performed, this testing will include a simulation of the TOR cryptosystem (key exchange, Cells Encryption/Decryption and Onion Wrapping-Encapsulation) into a JAVA platform. All the algorithms and mechanisms will be implemented in JAVA and tested. After confirming the elected candidate to be integrated into TOR for real-world simulation and testing, the candidates (Cipher Modes, New onion Construction approach) will be incorporated into the simulation platform TOR as new version of TOR running on the same simulation network to produce comparable results.

The proposed improvements will be included gradually and tested into the test-bed in the following planned order:

- Implementing, testing and comparing the obtained results of AES-OCB, AES-GCM and AES-CCM,
- Implementing the proposed Wrapping approach (multi-layered encapsulation), testing and comparing the obtained results,
- Implementing the two variant of Circuit construction algorithms, testing and comparing the results,
- Implementing the variable circuit length algorithm, testing and comparing the obtained results,
- Implementing the Multi-path Cell routing algorithm, testing and comparing the obtained results,
- Validation of the results and discussion.

In this chapter, a genuine and scientific assessment of the proposed improvement impact on TOR will be performed by implementing the proposed improvement into a testing platform and run the tests and comparing performances of both current and improved TOR.

The conception and design of a specific test-bed allowing the implementation and the test of the proposed improvement in a native, similar and accurate TOR environment along with avoiding any unethical activity by testing these improvement in the real world TOR. Therefore, this research will adopt a Model Net network emulation platform which will lodge a TOR specific simulation system called ExperimenTOR. ModelNet allows to implement realistic network structure, topologies, routing approach, delay and other features. Furthermore, it enables running real or modified C++ code (TOR) on the emulated platform to measure performances. The sought goals behind adopting this platform is to test the proposed improvement in a truly consistent, significant and reliable test-bed guaranteeing the same features as the real world TOR network in one hand, and on the other hand follow the ethical code and procedures which is crucial in this kind of research as adopting the real TOR with its users as testing platform will be considerate as unethical in the best-case scenario or illegal in the worst. Moreover, the measurement issue is also a preponderant factor on adopting this test-bed. In fact the implementation and testing phase seek to transit from theory to implementation and evaluation in practice, the sought measurements behind are to accurately

measure the timing, the resilience and the efficiency along with the compatibility and behaviour of the proposed improvement.

5.2. ExperimentTOR platform

To perform real and accurate experimentation on TOR without affecting other users, this work will implement the proposed improvements and enhancements into a TOR simulation dedicated platform called "ExperimentTOR" developed by Bauer et al. (2011) for TOR research purposes. It is based on the Model Net network emulation test-bed and can be used locally or virtually but requires several FreeBSD machines to work as ORs along with emulator core server responsible for networking the TOR architecture (Bauer & Sherr, 2012). Client (OP) can be run on several emulators. ExperimentTOR run the original TOR network deployment and code and can simulate different size network varying from 2 to several hundred depending on the componential power and memory res resources. ExperimentTOR is used to generate a downscaled network of TOR (small TOR prototype) and incorporate module for measuring and comparing TOR performances. Several reputed research on TOR used ExperimentTOR for emulation and simulation, such as (Wacek et al., 2013) research which relied on ExperimentTOR to perform an analysis of the OR selection technique and impact on anonymity and performance properties (Bauer & Sherr, 2011).

Moreover, (AlSabah et al., 2015) research used ExperimentTOR for evaluating their research on improving and surveying security in TOR. ExperimentTOR prototypes were deployed at many research centres in the United States and Canada. The current version consist of several FreeBSD machine with the ModelNet emulator kernel module and other machines running Linux working as ORs, OPs (clients), and application processes within the emulated topology. ExperimentTOR is a combination of python and C++ codes and include experiment and performances measurement toolkits such as "torperf" written in C++ for performance measurement purpose (Bauer & Sherr, 2011). For our work, we adopted this virtual test-bed which will be running on VIRTUAL BOX machine with the following architecture:

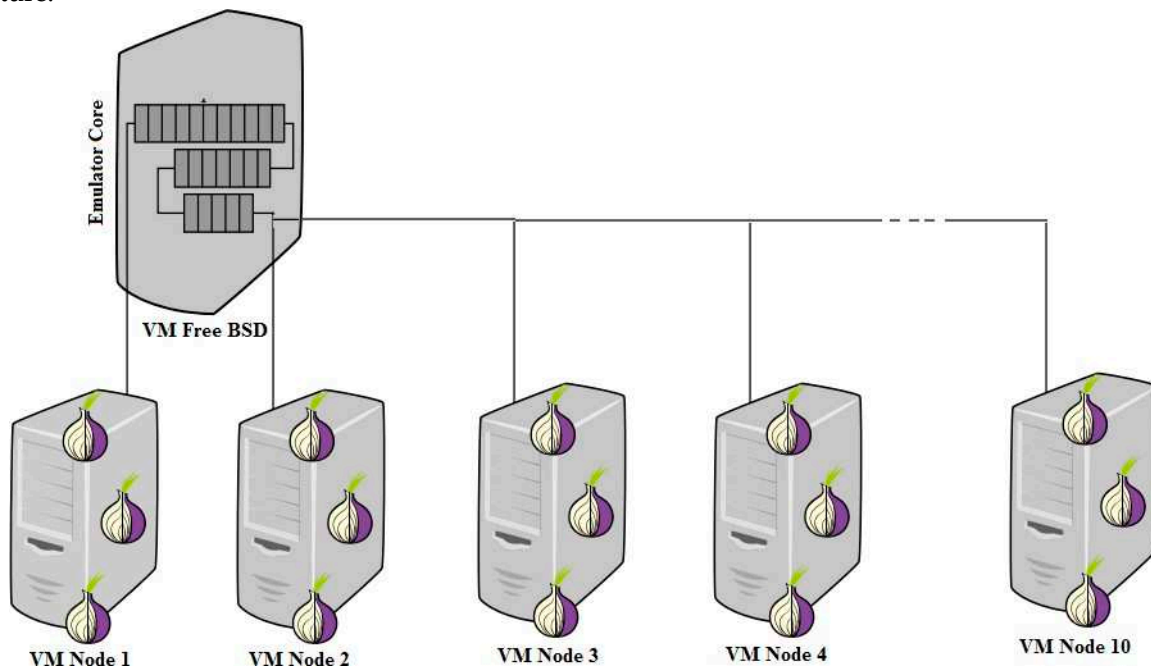


Figure 31. ExperimentTOR emulation platform architecture.

5.3. TOR test-bed topology and deployment

The first step into implementing the proposed solution and before running experiments in the test-bed is to design a TOR network realistic topology to be diploid later into the emulator. This work uses information from a real TOR Directory Server (DA) and applied the deployment into a small size simulation network at the corresponding componential power and bandwidth capacity of the ORs, Host and Server within the virtual topology by assigning a real bandwidth value and realistic network latencies to each end-host. In this work, the test-bed topology is as follow:

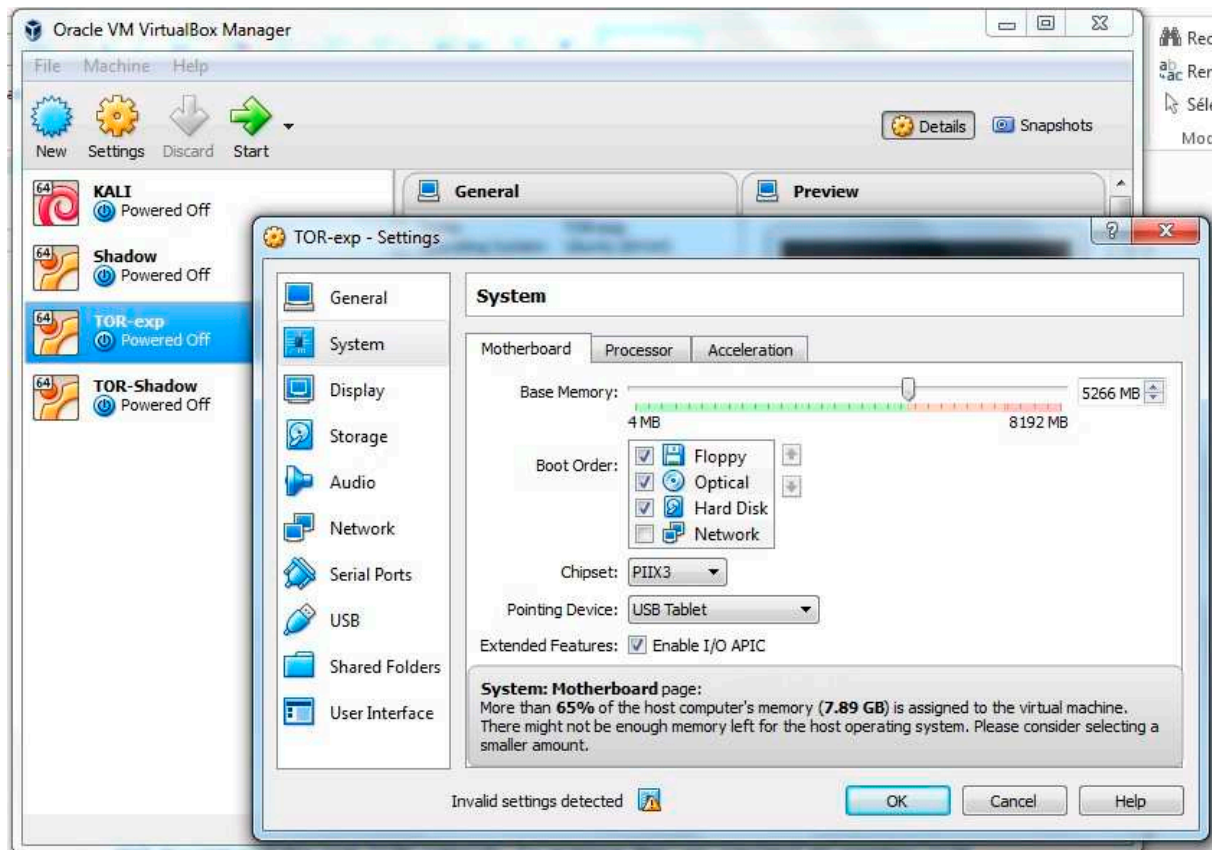


Figure 32. the simulation virtual environment configuration.

5.3.1. Simulation network size

The number of ORs into the simulation network used for implementation and testing is a crucial, because a small network might not reflect the same deployment or capture the same network features and effects that the real TOR network can experience. Moreover, the Network parameters should be adjustable to reflect the real-world situation and being flexible to allow the gradual incorporation of the proposed improvements along with the performance change testing and measurement. Moreover the virtualization option which this research adopt is tricky, despite the fact that this option will help to avoid compromising legitimate used and unethical, but also to run experimentations is a controllable and scalable environment, but the virtualisation will affect the performance and therefore the results of the experimentation. Thus, for each improvement incorporation test during this work, a benchmarking test will precede it using the current TOR deployment and feature which will serve as a comparison reference. The testing platform include the following:

- Twenty (20) FreeBSD Onion Routers including 3 dedicated Guard (Entry) ORs,
- One Directory Server (DA),
- Three (03) Onion Proxy (Clients),
- One service server (destination),
- Different size testing data

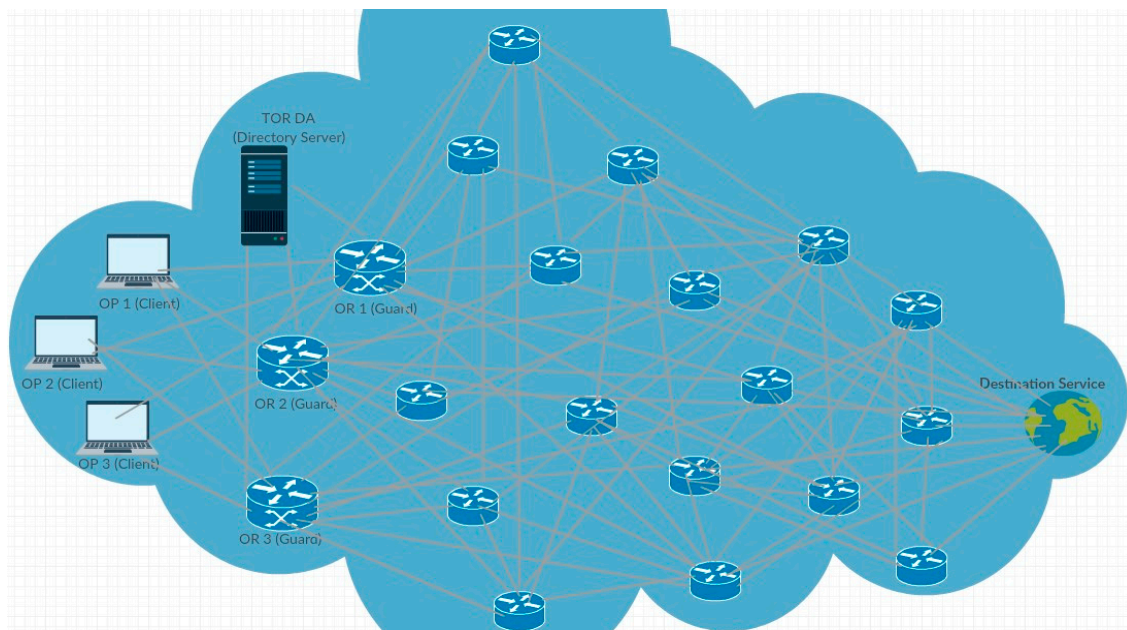


Figure 33. the simulation TOR network design.

5.3.2. Test-bed Preparations and configuration:

Before the actual tests could be carried out, the ExperimentTOR platform require reconfigured and benchmarking. Then, the proposed improvement were implemented in C++ and incorporated into the different modules of the test-bed. Moreover, some modifications were introduced to the modules: **torperf**: the performance measurement module in which several code addition and modification will occur in this module to include new functions required for testing purposes.

code: several modification and new implementation will be incorporated especially into the “crypto function”, “test”, “config” and “or” source code. The proposed cryptographic and onion construction improvement were implemented in this module.

log: new logging and processing function will be included in this module.


lighttpd: This is the source code for a very light-weight web server. A customised web servers to host web objects for clients to download will be included.

routers: Each router is assigned new routing, capabilities and type information along with numbered data directory. Routers 1-5 are configured as Entry (guard), router 18-20 are configured as Exit and all other routers are middle. The proposed circuit construction algorithm (ORs selection and routing approach, multi path routing and dynamic circuit length).

tcpping: use ICMP pings to measure round-trip times between virtual ORs. We introduce a new TCP- based ping timing measurement along with TCP's SYN and RST packets to calculate RTT.

tools: several script modification are introduced especially the configuring, running, and stopping functions.

torperf: the TOR performance assessor module which will be modified to measure several new variable related to encryption, circuit construction and routing approach performances.



```

INSTALL x README x crypto.c x
#include <stdio.h>
#endif
#ifdef HAVE_SYS_FCNTL_H
#include <sys/fcntl.h>
#endif

#define CRYPTO_PRIVATE
#include "crypto.h"
#include "../common/torlog.h"
#include "aes.h"
#include "aes-CBC-MAC.h"
#include "aes-OCB.h"
#include "../common/util.h"
#include "container.h"
#include "compat.h"

#if OPENSSL_VERSION_NUMBER < 0x00907000L
#error "We require OpenSSL >= 0.9.7"
#endif

#include <openssl/engine.h>

#ifdef ANDROID
/* Android's OpenSSL seems to have removed all of its Engine support. */
#define DISABLE_ENGINES
#endif

```

Figure 34. example of code modification within the adopted simulation platform.

5.3.3. Logging and measurements

Native TOR implementation provides several measuring and logging features varying from a high-level error logs to a detailed performances measurement. However, these log still present crucial lack as the information does not include performance logging.

In this work, the logging for tests and experiences is ensured by the embedded ExperimentTOR logging module called "TORPerf". Though, several modification were introduced to deal with the edited module and function. Logging is performed at the network level (directory server), ORs level and at OP level (TOR software).

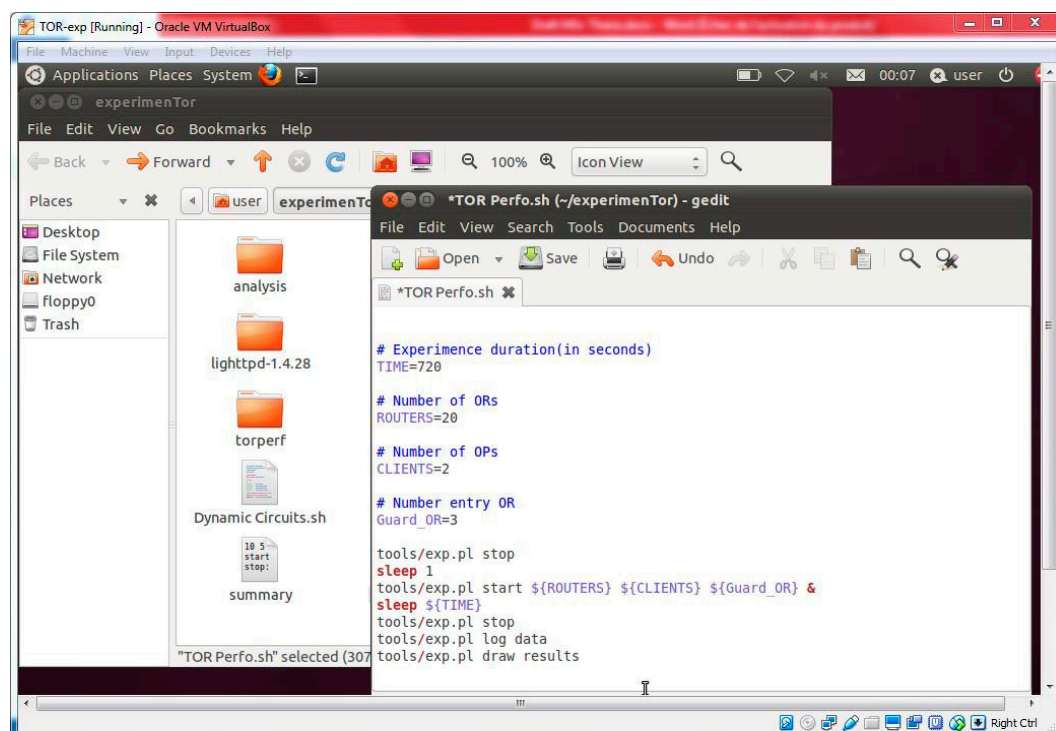


Figure 35. example of experiment setup and configuration in ExperimentTOR.

5.4. Testing and results

In this section, the testing of the implemented proposed improvements and enhancements will be performed several times and results will be logged in details

Finally, after testing the proposed wrapping (onion construction) approach. The algorithm is implemented in Python (plus the Browser C++) and integrated in the TOR simulation platform (TOR browser and the ORs) to be tested in TOR condition, measure the performance enhancement and assess the suitability.

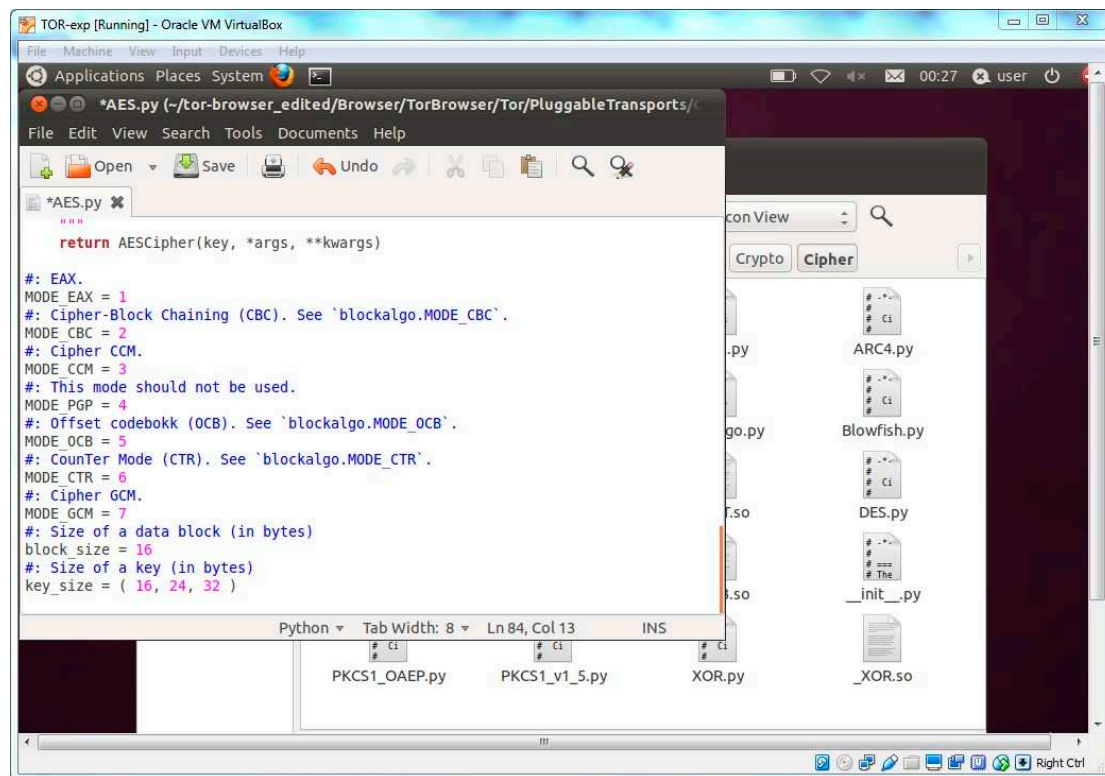


Figure 37. the implementation of the selected modes of AES in the ExperimenTOR platform.

a) Experimentation Results and Discussion

The results obtained initially Java implementation of AES encryption algorithm in three modes; CBC (first generation TOR), CTR (second generation) and OCB (with disabling the authentication TAG processing and only performing encryption) in order to pre-assess the efficiency of each mode for different data size varying from 16 to 5120 bytes. The results obtained were predicted and showed that CTR (without authentication) perform better than OCB and CBC. Thus, without the need of node-to- node authentication the current TOR adopted mode seems to be the perfect choice. The figure summarize the performance of each mode.

Table 2. the AES modes testing results on JAVA.

Input size (Bytes)	16	64	128	256	512	1024	2048	5120
Time Consumed (ms) per mode								
CBC	64.11	102.31	198.34	344.76	634.21	1123.44	2011.54	4329.33
OCB	10.54	34.25	56.23	98.50	176.32	322.90	603.74	1488.60
CTR	5.82	20.58	36.77	65.88	129.98	245.34	468.13	1198.86

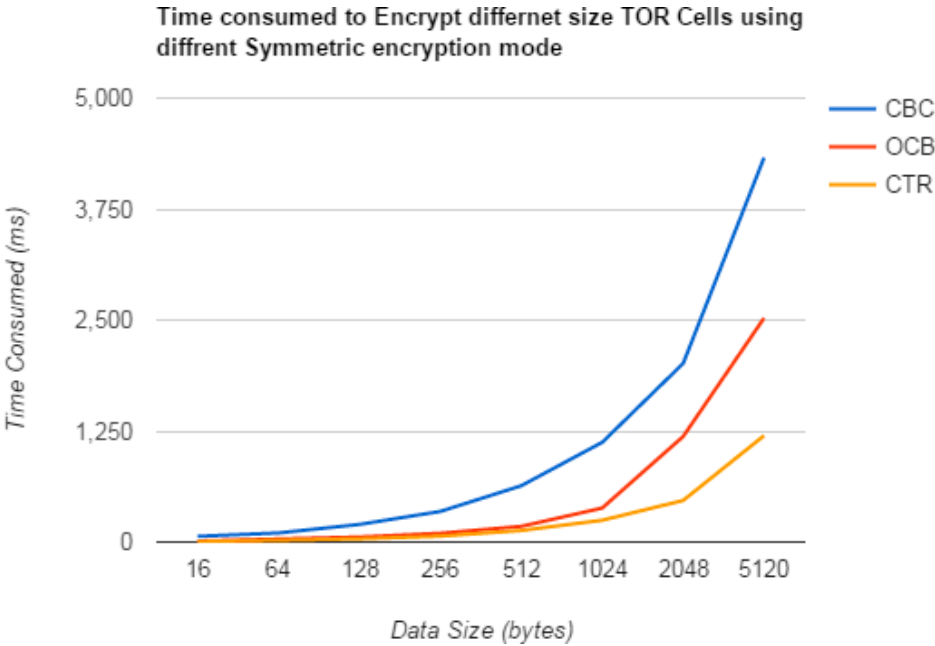


Figure 38. the comparative graph of the AES encryption only modes performances.

Then, we implemented (coded) an authentication processing function to CBC and CTR mode with preserving the same encryption implementation, where the OCB was implemented entirely by activating the authentication function. We tested the three implementation in the same way. Here, the results obtained (figure) were different. Regardless the slightly slowdown in OCB performance, it perform better then CBC+MAC and CTR+MAC. Therefore, to introduce the node-to-node authentication in TOR it is obvious that the classic implementation of encryption followed by authentication is a wrong decision.

Table 3. the performances of AES modes encryption plus authentication.

Data Size	16	64	128	256	512	1024	2048	5120
Time Consumed (ms) per mode								
CBC+MAC	84.11	122.31	218.34	364.76	674.21	1323.44	2191.54	4629.33
CTR+MAC	35.82	80.58	96.77	165.88	329.98	645.34	1068.13	2398.86
OCB	29.54	54.25	76.23	138.50	246.32	472.90	793.74	1578.60

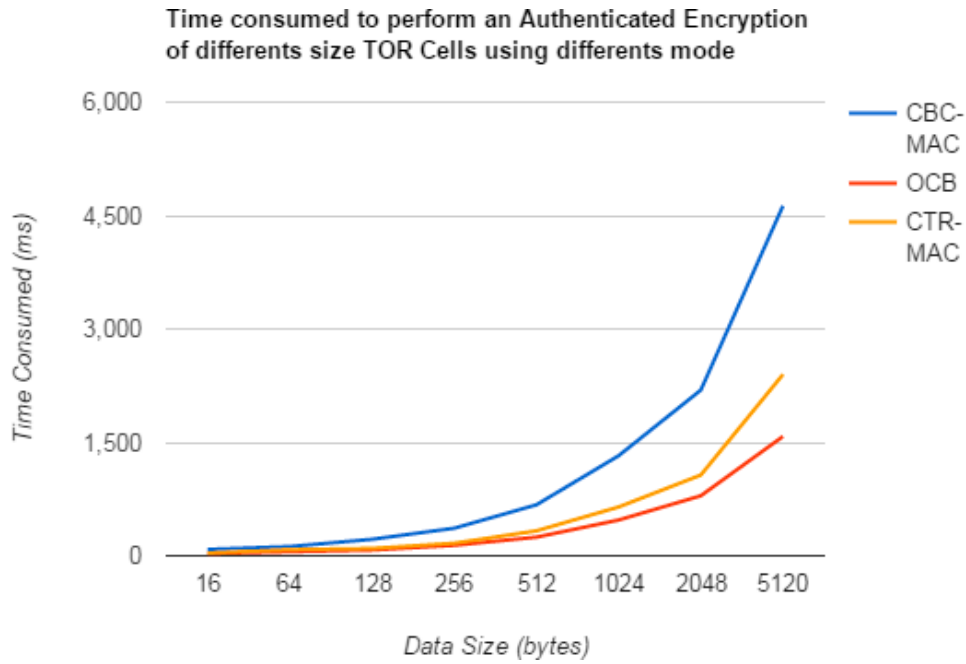


Figure 39. the comparative graph of the AES authenticated encryption modes performances.

Nevertheless, OCB is not the only AES implementation mode guaranteeing an integrated authentication. Three other modes; CCM, GCM and EAX should also be considered. Therefore, we implemented the six (06) mode together using the test-bed resources (javax.crypto.Cipher) in which the implementation is optimised and implemented natively. And the results obtained (figure) showed that three candidate could be replacement of the current TOR mode; CCM, GCM and OCB.

Table 4. the performance of all AE candidate in java compiler.

Data size (Bytes)	16	20	32	64	128	256	512	768	1024	1536	2048
Time consumed (ms) per Mode											
CBC	136	167	207	232	248	259	269	278	286	299	312
CTR	89	109	140	162	193	225	246	261	269	275	282
EAX	91	98	123	133	162	177	193	216	228	257	268
CCM	52	59	79	83	102	129	148	157	160	167	174
GCM	45	51	73	75	88	104	116	127	131	139	153
OCB	36	48	65	73	89	101	113	120	127	135	148

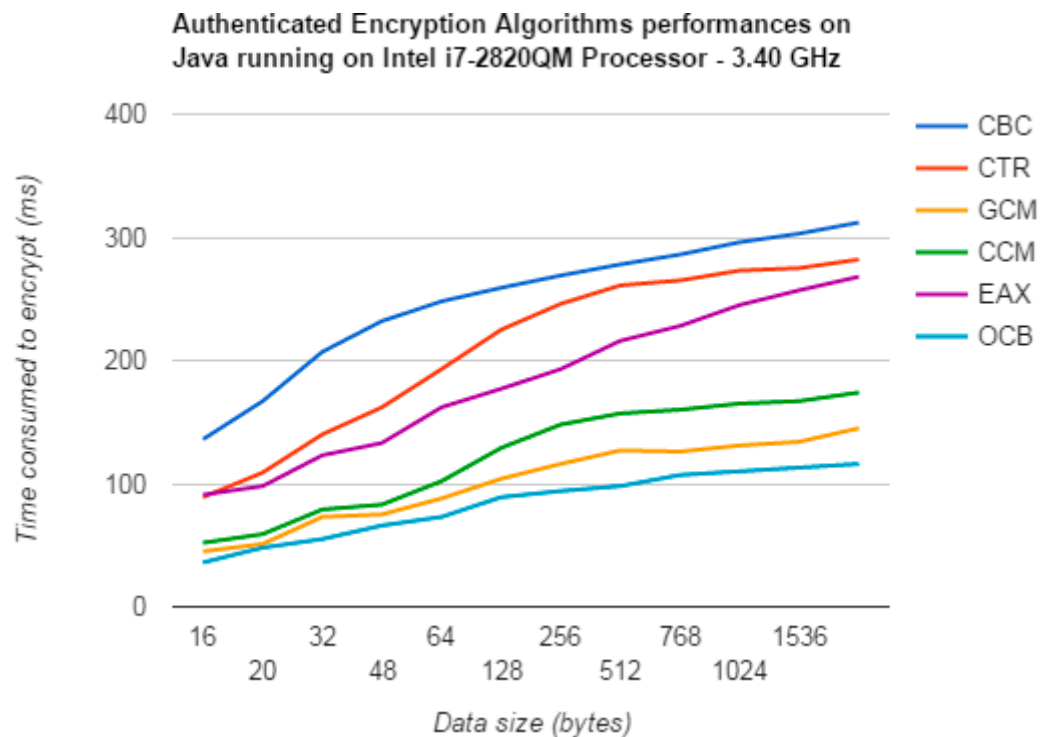


Figure 40. graphic representation of the performances of all candidates inn JAVA.

b) Encryption testing results

From the obtained results and the research carried on the authenticated-encryption, OCB is by far the best integrated mode as it perform both encryption and authentication in one round (single pass). OCB is patented in US in case of commercial use, however TOR is an open-source and free software/service and thus this restriction doesn't apply. Moreover, OCB accomplishes the authentication processing without using the heavy-weight universal hashing computing which make it not only the best mode in term of performance but also in term of resources use and the implementation in both hardware and software. Nevertheless, the Nonce generation choice and use in online system can be tricky as Nonce require to be unique for every encryption (no need to be random as it is the case of IV in CTR, GMC and CCM modes).

c) Resources use analysis

The Performance comparison is not enough to constitute a criteria for selecting an encryption algorithm or mode, in fact the componential power required/consumed by a cipher is also an important factor. In TOR context this power is limited and therefore precious. In term of Memory requirement, OCB has the smallest memory usage as it mainly relies on the block cipher operation and data with no use of other function. The memory required for GCM is the largest due to the Galois function and it uses more extra pre-computation table for speed acceleration. On the other hand, the computation cost of CCM is the highest compared with the OCB and GCM as it requires two cipher invocations for each block, while OCB mode require merely one cipher call for each block along with the special function which have a low computation complexity.

d) Choice and Discussion

In the real world situation, encryption modes present various pitfalls due to the combined privacy and authenticity processing such as the failure of guarantee a proper key separation, the misuse of the MAC which open security breach into the cipher allowing cryptanalysis or brute-force attacks, mismanaging the IVs or Nonces. OCB and GCM are provably secure meet these required standards but remain dependant on the correct implementation especially on software which mean more vulnerabilities threats and enemies. On this work a high importance is accorded to mode selection, thus we tested all the potential authenticated-encryption schemes which are OCB, CCM and GCM. The tests conclusions are as follow:

- **CCM** is inspired from the generic composition of CBC+MAC mode allowing a moderate performance by the use of integrated authentication scheme and optimize the implementation. However, it remains less efficient in term of performance than GCM and OCB,
- **GCM** is an improved incorporation of CTR mode encryption mechanism with an integrated internal MAC allowing parallelizable operation and thus a better performances and resources efficiency. Nevertheless, GCM parallelization feature have a significant impact only on high-speed hardware-based applications. In TOR context, because the cryptographic operation are performed on software basis the GCM mode performance was not optimal and hence not as well as OCB.

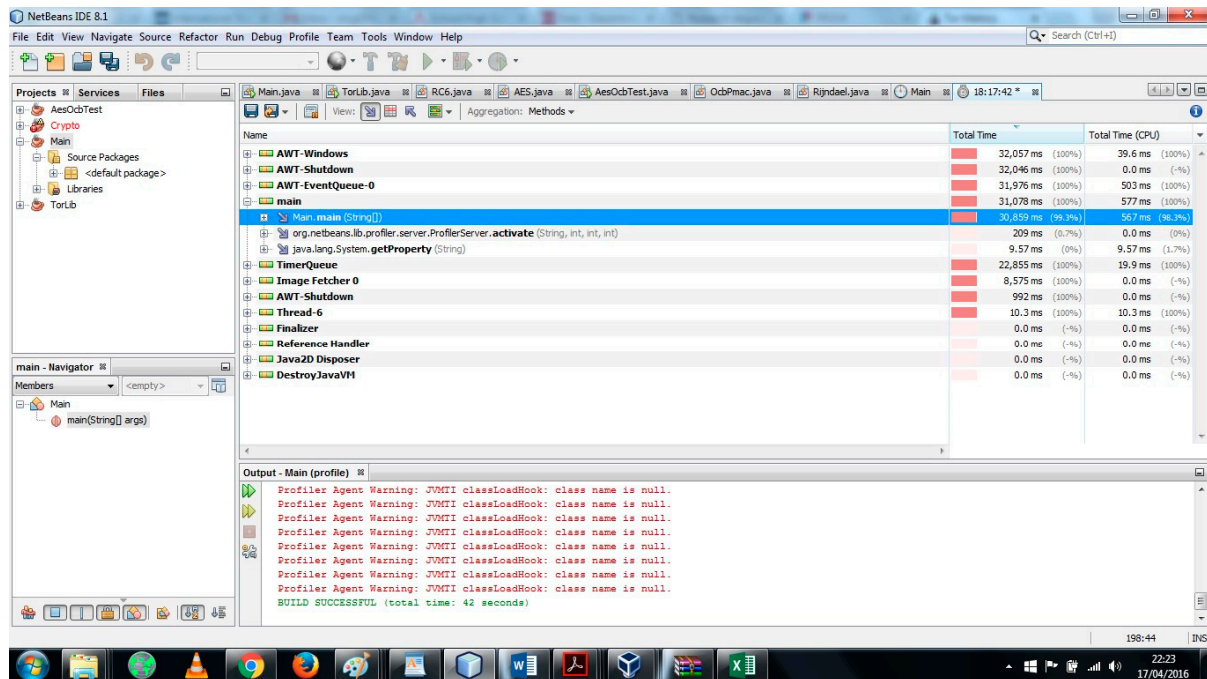


Figure 41. performance and timing measurement example on JAVA platform.

5.4.2. The Onion construction (Encapsulation)

The implementation of the proposed encapsulation approach was implemented on the JAVA test-bed in a light-weight way by just including the cell construction, headers incorporation and dis-association, public key encryption symmetric keys exchanges, onion encryption. For testing purposes, this work eliminate system I/O and reading function use and was restricted to the minimum required. The results obtained for the proposed approach (figure) shows the clear performance enhancement brought by the proposed encapsulation approach. In fact, the proposed approach save two AES encryption round of data size equal to the initial data size (intended to be sent through TOR) which reduce significantly not only the time required for wrapping but also the un-wrapping (decryption) at ORs level. Therefore, the performance enhancement is not limited to the onion construction but extended to alighting the routing through TOR.

a) *Obtained results:*

The testing of the proposed onion construction approach produced the following results:

Table 5. construction programs performances in JAVA platform.

Cell Size (Bytes)	512	1024	2048	5120
Current TOR Cryptosystem (ms)	834.81	1463.65	2390.03	4388.41
Proposed Cryptosystem (ms)	521.90	877.74	1730.11	2912.43

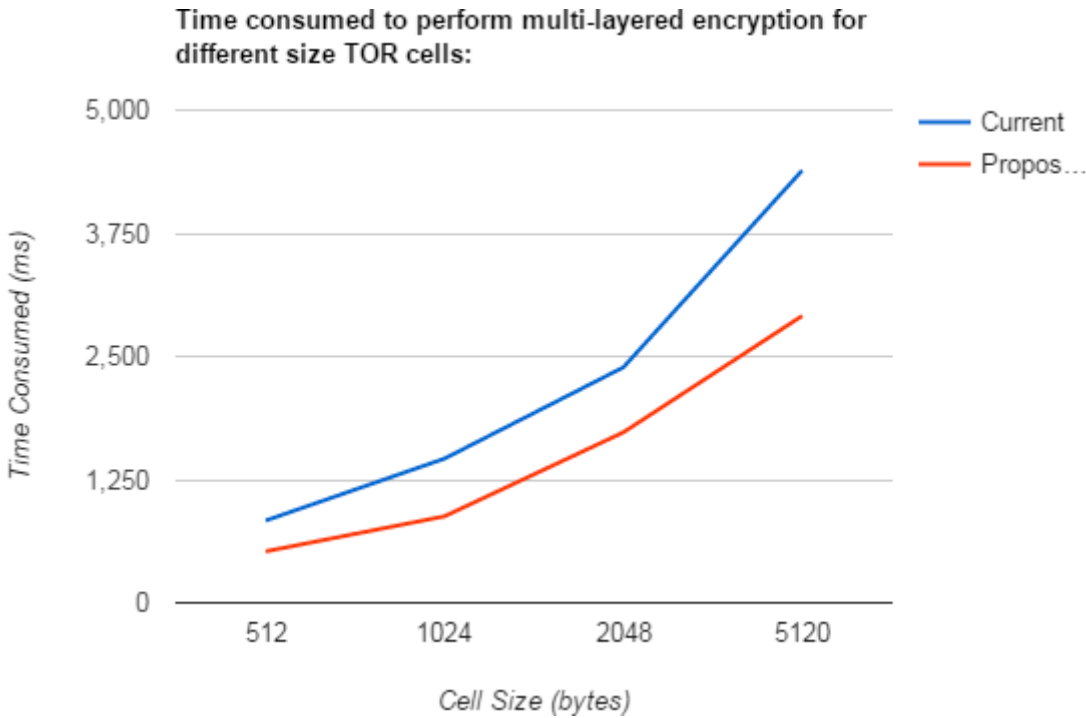


Figure 42. Time consumed to perform multi-layered encryption of TOR cells.

Moreover, the function of measuring the operations carried out during the execution of the cryptographic operations within the TOR-like cryptosystem produced the following results:

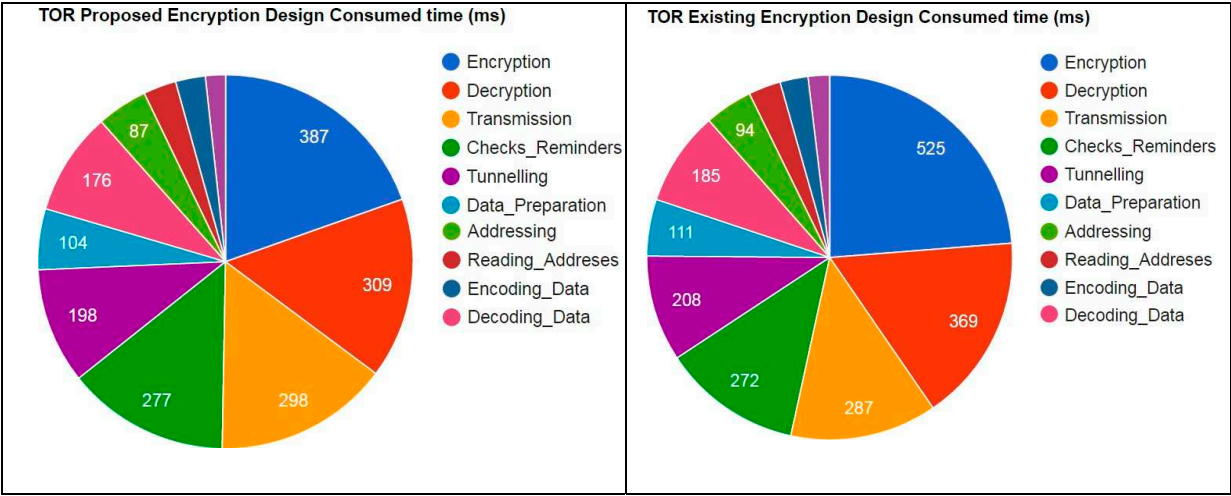


Figure 43. the time consumed per task on TOR cryptosystem.

From the two above pie-charts it is clear and obvious that the proposed approach reduced considerably the time spent on Encryption and Decryption and also induced a slight improvement (time consumed reduction) for other operations such as encoding/ decoding. However, slight increase on the time consumed on checking, transmission and addressing was observed, nevertheless this small changes do not impact the overall performances improvement.

Finally, after confirming the efficiency of the proposed encryption mode. An implementation of AES-OCB was performed and replaced the current implementation into the TOR simulations platform (ExperimentOR) has produced the following results (figure). Moreover, an implementation of AES- CTR mode with a separate authentication (MAC) computation was performed for comparison purposes.

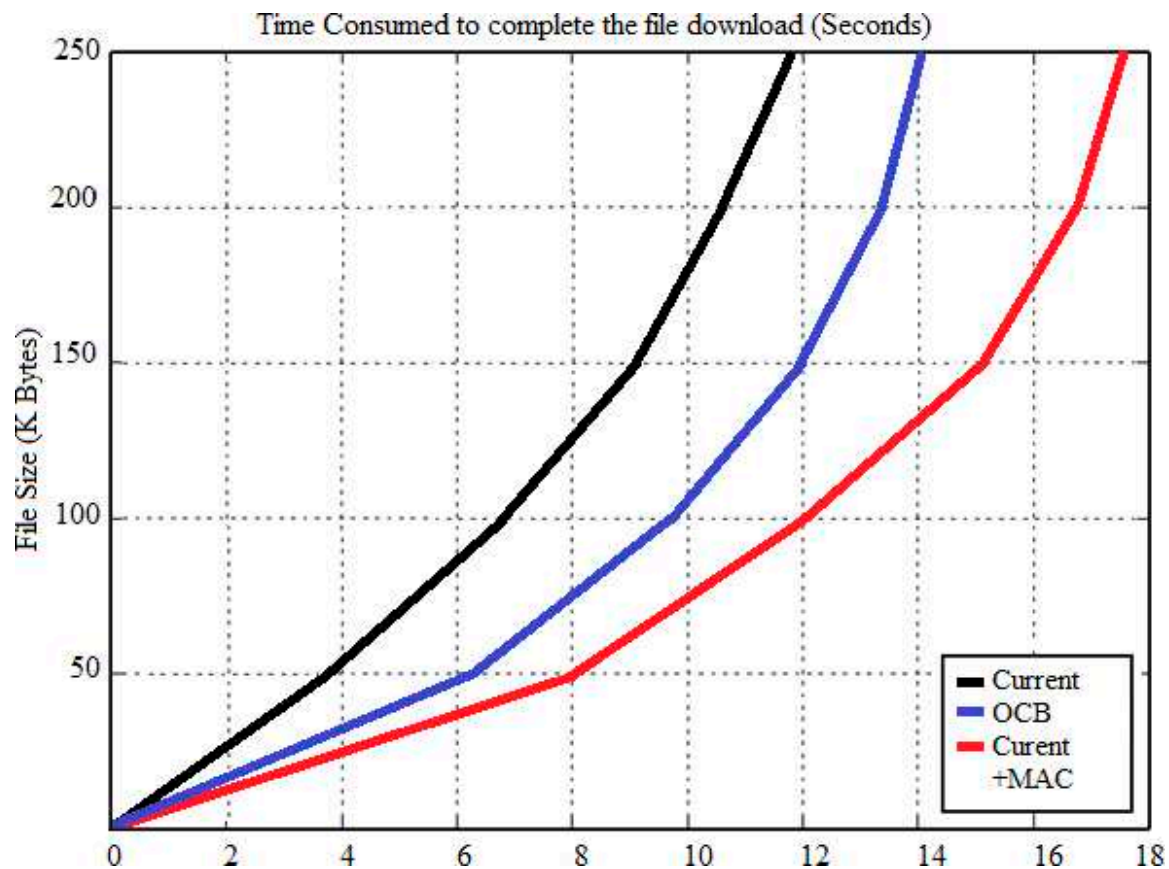


Figure 44. AES encryption modes performances on ExperimenTOR.

b) *Discussion of the obtained results:*

The two pie charts provide an overview of the performance enhancement brought by the proposed approach. In fact, to better explain and distinguish the different results we categorized the TOR cryptosystem mechanism into function and measured the CPU time consumed to complete each stage. Where the encryption and decryption seems to consume almost the half of the cryptosystem time, the proposed approach reduce this time significantly by around 23.5%. Furthermore, the remaining functions performances (transmission, checks, encoding, decoding, data preparation ...etc.) were also slightly enhanced. In fact, regardless the importance of these functions for TOR, their impact on the overall performance is huge and it can be can observed that the encoding, decoding, addressing and check takes a lot of CPU time but no further improvement is unfortunately possible.

Despite the fact that the introduction of OCB mode will slightly affect the performances of TOR, the security assured by the authentication is crucial and therefore the OCB is the perfect balance between security and performances.

5.4.3. ORs Selection and Circuit Construction testing

In this work we implemented the algorithm of circuit selection into the ExperimenTOR system, the program is written in C++ within the TOR path simulator program. The code receive as input an existing consensus from the DA and generates circuits with specified constraints, such as the list of ORs and types, performances. The program compute the probability that each ORs part of the selected circuit is compromised and measures the time consumed to construct the circuit and the performance during the download of fixed size data. We compared two circuit (path) selection algorithms implemented into the ExperimenTOR in addition of the existing TOR circuit selection algorithm which will be used as a reference for evaluating the proposed improvements, the two implemented algorithm are:

- Bandwidth-Weighted algorithm optimised by a combination of Dekjstra and A* algorithms,
- Controlled Exit OR Algorithm with random intermediate selection.

The simulator generated several hundred of circuits (for both algorithm) and transmitted different data size throughout along with measuring the performances with the probability that fake (malicious) ORs are being part of the circuit in order to truly evaluating the possibility of successful FBI attack. Moreover, a specific attack evaluation

algorithm was adopted (Steven et al., 2011) to determine and measurement the impact of misleading (false) bandwidth information share by the ORs.

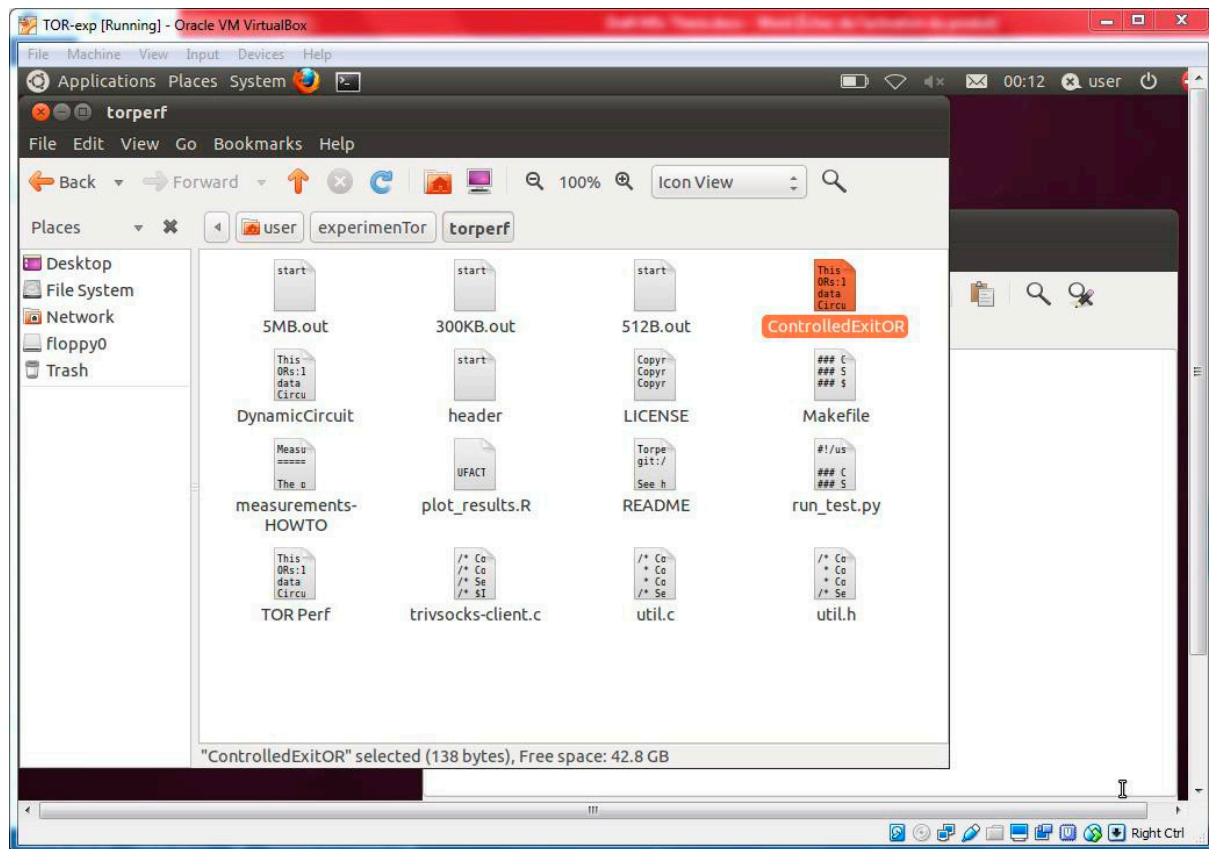


Figure 45. circuit selection and routing algorithm implementation in ExperimentTOR.

c) *Obtained results:*

In the first experiment, shown in Figure 1, where variant number of compromised ORs were injected into the simulation network and the measurement function determined the percentage of the occurrence of a successful FBI attack on the three algorithms. The results shows that the current TOR uniform ORs selection and the other TOR variant relying on capacity in assigning ORs leaves a high probability for an attacker who compromised a certain number of ORs to fully control the circuit (FBI Attack). Nevertheless, the current number of ORs in use into the real world TOR is about 6000 and the results of a simulation into the real world could lead to different results. On the other hand, the proposed circuit construction approach relying on the selection of the Exit ORs from a trusted (pre-defined) set of ORs shows its efficiency on dealing with such attacks. In fact, this approach guarantee that the selected Entry and Exit ORs are safe and prevent an attacker from influencing the process of selection by injecting powerful ORs which provide false and misleading information to the TOR Directory Authority (DA).

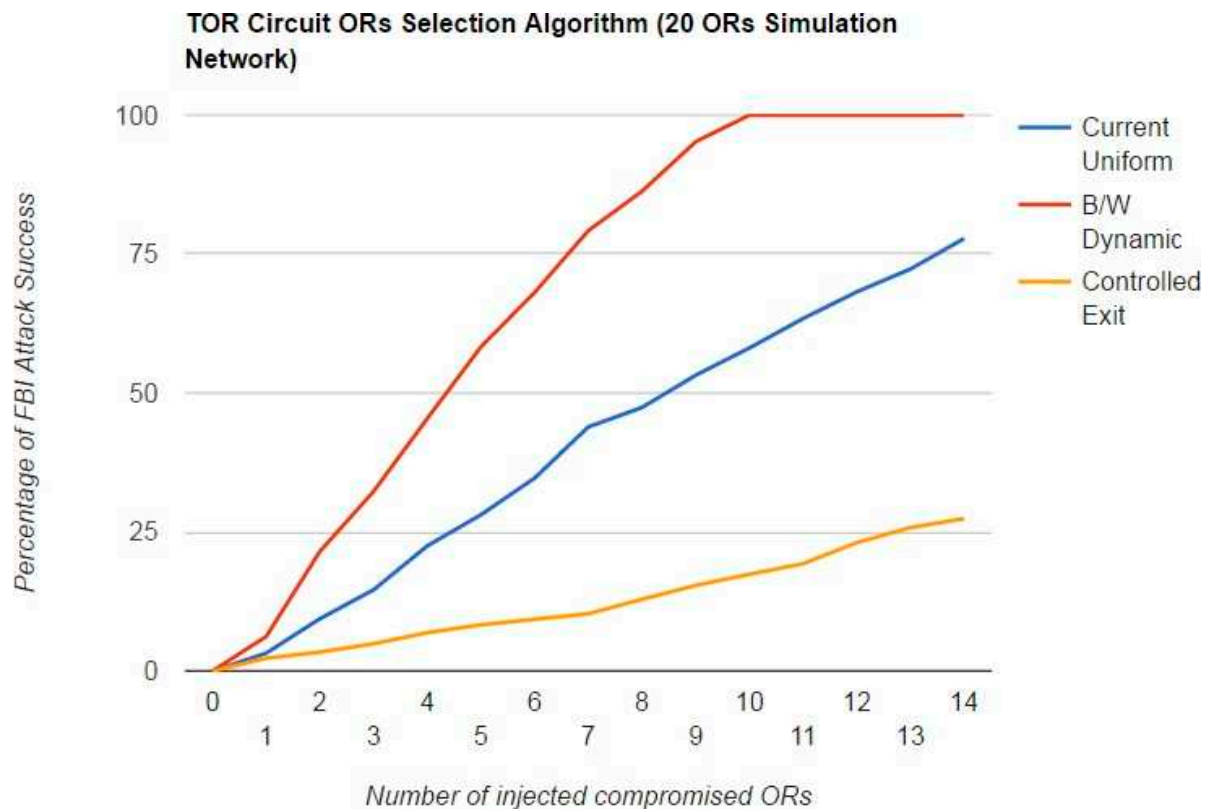


Figure 46. FBI attack occurrence probability by number of injected compromised ORs.

d) **Results discussion:**

The analysis of the effectiveness of the different TOR circuit selection algorithms in term of performance and security which are in reality closely related as the TOR performance affect the security because many attack exploit the delays and the compromise made by TOR developer to maintain the required performance and thus the system usability. The following graph (Figure 47) illustrates the use of available bandwidth for each algorithm. In fact, despite the security enhancement brought by the controlled exit circuit construction approach, the impact on performance is significantly negative as the ORs will not be able to exploit at maximum the available bandwidth and thus reduce the performance. While the dynamic circuit construction seems to be the best in term of performance, this approach is risky as any attacker who possess the adequate resources can inject powerful compromised ORs and attract more traffic through these ORs and therefore compromise user anonymity. This attack was already carried out by researcher working for the FBI in 2015 and achieved its goals by de-anonymise users.

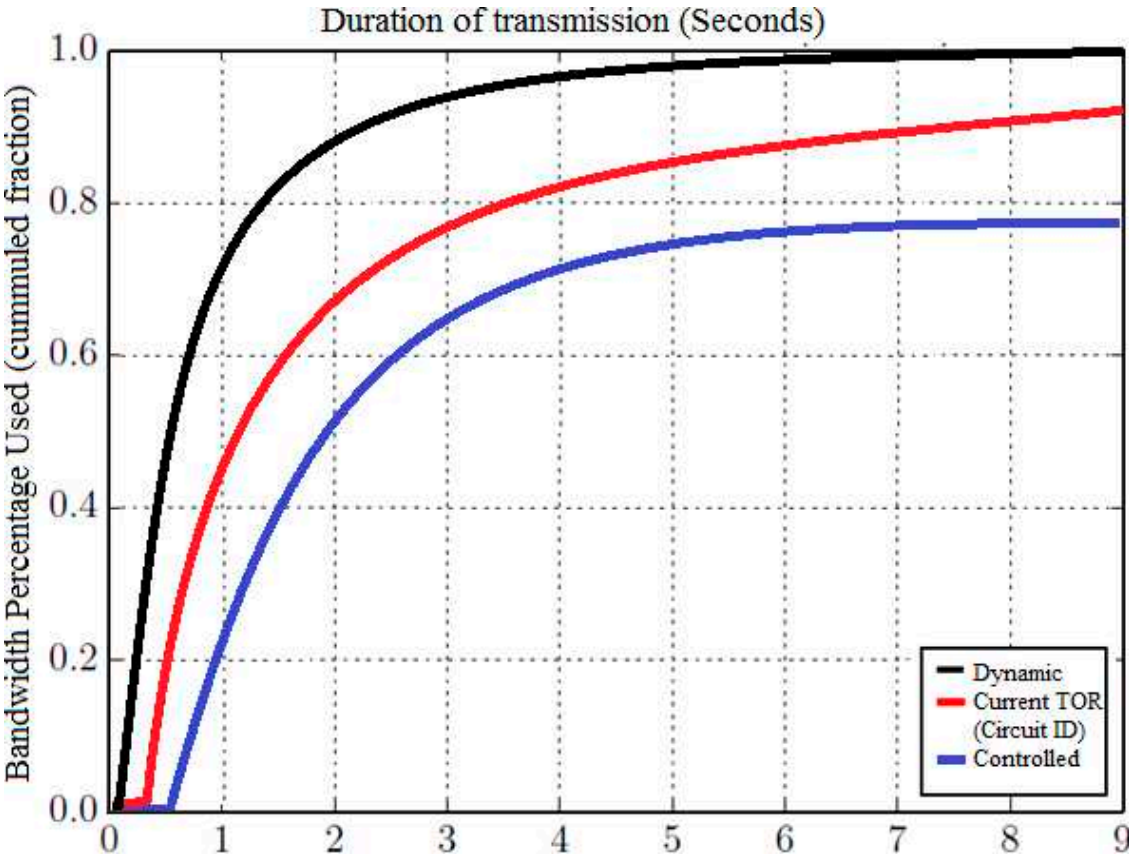


Figure 47. Circuit construction algorithms bandwidth usage on ExperimenTOR.

5.4.4. The Multi-circuit routing testing and results

The implementation of the multi-circuit routing function within the existing TOR routing approach (circuit ID) and the Proposed Approach (controlled Exit) produced unexpected results. The proposed approach performed better in the multi-circuit routing approach then the existing approach (figure). In fact, the multi-circuit approach registered les 22% in delays then the concurrent when the current TOR single circuit per session remains better in term of performance but vulnerable in term of security. Therefore, the proposed approach could be considered as the best balance security- performance. It is also recommended that the TOR incorporate the proposed approach along with preserving the existing one.

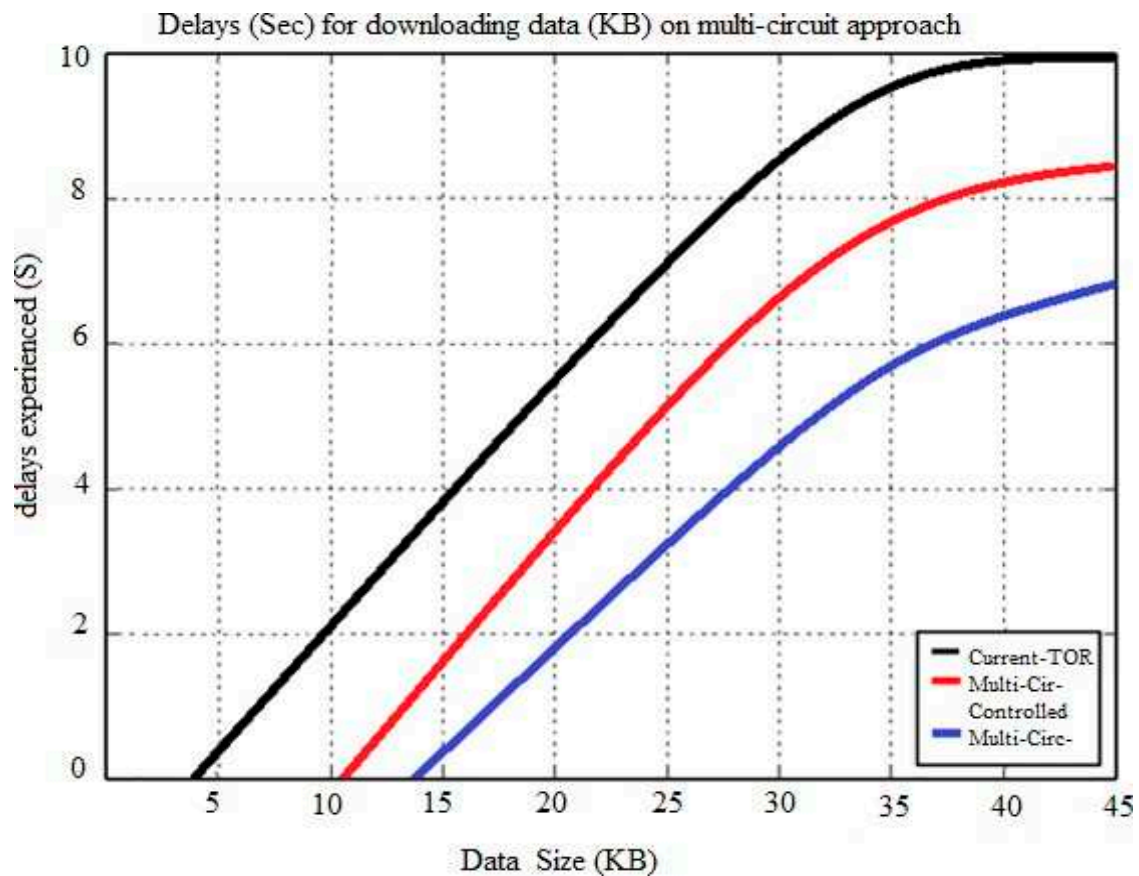


Figure 48. Delays during downloading data per algorithm.

To summarize, the suitability of the proposed improvement are discussed on two major factors; the security and the performance. When it seems obvious that some proposed improvement could cause more performance slowdown on TOR, the security enhancement and enforcement which these improvement provide is significant. Nevertheless, the proposed improvement in routing and circuit selection have brought some performance enhancement and therefore, the TOR should allow the user the choice between better security or better performances.

5.4.5. The dynamic circuit length results and discussion

The variant length circuit algorithm was implemented into the ExperimentTOR platform and tested in two scenario to produce the following results:

Table 6. The TOR performance in downloading for variable circuit length.

Number of ORs in the circuit	0	1	2	3	4	5
TOR current deployment performance (KB/S)	65.5	39.1	25.3	22.1	19.7	17.4
Proposed enhancement performance (KB/S)	65.5	50.2	39.8	32.9	27.2	25.5

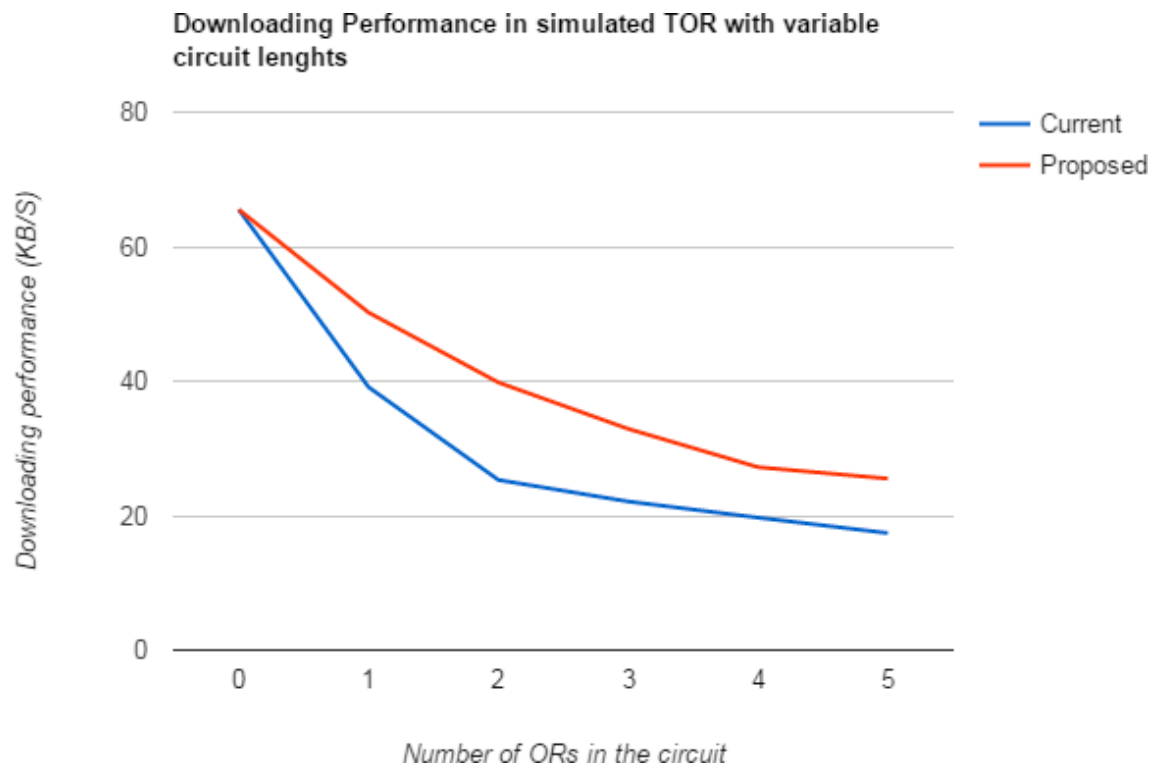


Figure 49. graph representing obtained results for dynamic length circuit usage.

The obtained results, illustrates the performance enhancement of the proposed approach and their impact also on the TOR security. In other word, the proposed improvement produced almost the same performance in a circuit of 4 ORs length as the current TOR implementation for 3 ORs. This improvement have a huge impact on security as the use of 4 ORs circuits will enhance considerably the TOR security and resistance to several attack (FBI, Timing analysis, path selection ...etc.) because the likelihood of selecting a malicious (compromised) OR in a specific position within the circuit remains the same. However, the possibility of choosing such OR in the given position into the circuit will decrease as the number of ORs in circuits increase and therefore the number of circuits. Thus, considering the proposed improvements together could strikes a good security-performance balance as the time gained by the proposed encapsulation approach could be later scarified for improving security by introducing the new circuits' construction and routing approach.

5.5. Results' validation and findings summary

This work has permitted to validate the suitability and evaluate the efficiency of some proposed improvement to be integrated into the real TOR network. In fact, implementing and testing the proposed performance improvement and security enhancement produced in the overall the expected results with some exception. The results obtained from testing the incorporation of authenticated-encryption mode into TOR symmetric encryption (multi-layer) shows that it will meet the two initial set goals; improving performances and enhance security. In other words, the OCB mode implementation behave as was expected and ensured the TOR node-to-node authenticity alongside with limiting the performances slowdown impact.

On the other hand, the proposed approach for onion construction implemented and tested on both test-bed has produced excellent performance improvement along with preserving the same security offered by the currently adopted approach. The cryptography behind the TOR does not constitute an exception in term of principle, and the redundancy employed in the existing approach has a severe impacts on TOR overall performance and in some situation security (some timing and profiling attacks are due to TOR delays). No crypt-analysis or brute force attack were attempted against the encryption as the used mode and algorithm are known to be provably secure (mathematical and componential security). The security assumption for the sufficiency of only one layer of encryption for cell data comes from cryptographic research outcomes (demonstrated), hence the proposed cryptographic solution are, to the best of our knowledge, secure and reliable to be implemented on the TOR. Note that the issues related to the Patten of OCB in US does not apply for TOR as it is open source and free solution.

Regarding the routing and circuit construction proposed solution, where some results have shown the efficiency and permitted to validate the persistence of certain proposed solution and confirm the unsuitability of others. The proposed circuit selection algorithm based on the controlled exit technique is undoubtedly less efficient than the existing algorithm, however if the security is considered and accorded the required importance the proposed solution is by far better. In fact the time saved by the proposed encapsulation approach could be in part sacrificed in the circuit selection process to guarantee a better security and resistance to attacks. Moreover, the proposed multi-path routing solution is also producing encouraging results and should be carefully considered. On the other hand, the variant length circuit technique have been proved to be limited and will, in case of adoption, impact significantly badly the overall performance without providing the required security.

An important point should be highlighted, as the security and ethics behaviour have pushed this work to be tested into virtual small scale TOR-like simulation network which could have a negative impact on the accuracy and the acceptance of the obtained results. Nevertheless, the test-bed were designed and adapted to better reflect a real world TOR network and thus provide the best possible results quality.

Chapter 6: Conclusion

6.1. Work output and conclusion

This work proposes several cryptographic and routing improvements for the second generation (current) TOR network. The proposed solution covers two main area; enforcing the security and enhancing performances. The authenticated-encryption AES mode called OCB and the new approach of onion construction (encapsulation) aim respectively to enforce TOR security by providing a crucial Node-to-Node authenticity and lightening the multi-layer encryption approach along with preserving the same security level. In fact, TOR cryptosystem present several useless redundancy which cause more delays with no security clear or proved enforcement. This work adopted a scientific approach into the proposition, implementation and testing of the proposed cryptographic improvements as it first started by investigating the different candidates and solution, then testing these candidates into a purely cryptographic environment, and finally implementing the selected solutions into a specific simulation and platform to assess the performances and compare them to the existing implementation. The adopted research, implementation and testing methodology consolidate the trust into the obtained results which showed the sought efficiency and the suitability.

On the other hand, this work targeted to improve the TOR security and performance by enhancing some non-cryptographic mechanisms (routing and functioning) which are closely related to the TOR cryptography, in fact the existing TOR circuit (routing path) selection and construction mechanism which seems to be a pure routing matter is very related to TOR security. Hence, this work proposed some improvement regarding the circuit selection approach (controlled exit) which could contribute into the TOR resistance especially against path selection attack (known as FBI attack) which was very effective into de-anonymise TOR users. Moreover, a multi-circuit routing approach was proposed which aims to balance the security and performance over TOR. Nevertheless, the proposed TOR dynamic circuit length solution testing showed its limits as it impact significantly the performance without a clear performance improvement.

The implemented improvements are based on previous academic works and research outcomes especially regarding the implementation of the third version of the OCB mode appeared in 2014. The emulation platform ExperimentTOR was slightly modified to include more measurement functions on one hand, and on the other hand accommodate the proposed improvement. Nevertheless the core of the platform remains the same.

6.2. Further works

Regarding possible future work in this topic, several possible research directions could be investigated. These are mainly related to improving and advancing the current TOR protocol, mechanism and defences. In fact, TOR enemies list is getting expanding every day and so is the need of online privacy and anonymity for people. First of all, a research project could work on the next generation of TOR in the post-quantum computing. Secondly, the IPv6 migration is a new challenge for TOR future and should be addressed properly. Thirdly, the current TOR overall design should be deeply reviewed to incorporate so revolutionary solution and move to virtualisation or cloud computing.

The most important research direction which should be considered is to investigate the possibility of auditing anonymously TOR user's activities to distinguish between genuine users aiming to protect themselves and criminal (all categories included) which are abusing the system. This research could solve one of the most problematic point that the TOR network is facing and throw away any criminal responsibility that some governments are sticking to TOR as a solution that help criminal to escape and evade prosecution.

6.3. *Work Evaluation*

Many methods, techniques and approaches were used during the development of this work. Some of them new but mostly the author of this Dissertation has applied the knowledge and skills gained in the Taught and developments modules of the two previous semesters of the MSc. course. In approaching project management, together with the techniques and tools used the author demonstrated his skills in this field. Giving the nature of the research carried out in this work, the application or adoption of the classical research and development methodologies. The [practical implementation and testing of the proposed improvement and enhancement was preceded by a scientific evaluation (theory) in which the proposed solution were studied in deep. During this work several detail were deliberately dropped as the amount of technical details could impact the meaning. Nevertheless, all details are presented in appendixes; AES algorithm details, TOR detailed specification and cryptography use, detailed functioning of some features.

This work constitutes a scientific contribution into the research of enhancing and improving TOR network security and performance. A deep investigation of the weaknesses and flaws causing the security issues and slow performance on current TOR implementation was carried out at the beginning of the work. Later, a scientific approach was adopted to validate the selection of the most suitable solution among the candidates. Finally, a specific test-beds were designed in which the selected solutions were implemented and the required modification were performed in order to assess in TOR- like environment the performances improvement and security enhancement brought by the proposed solution and therefore validate the findings.

6.4. *Reflective and Critical Evaluation*

The accomplishment of this work has been a challenging task for me. Under the self-established standard and the pressure imposed by myself for passing this module with distinction, this work seemed to be insurmountable. Knowing my capabilities, I was sure from the beginning of this work, looking at my performance in the taught semesters and my academic background, that there will be no reason I would not to be successful in this last task.

Probably the biggest concern was the complexity of the topic that I had chosen. Starting with a limited knowledge in C++ and Python (usually program in JAVA) and not having a clear idea about the platform for performing simulation and testing and therefore the risk was relatively high. In addition, I had decided to tackle several points and not only limited to one improvement. Moreover, the respect of ethical and professional standard required to carry out this work had led me to adopt a virtual simulation platform rather than using it on the real world TOR and therefore breach the rules and out the users in risk of disclose, this step was the crucial one as the used simulation platform required a lot of modification and coding (add new modules) assessing and performance measuring functions, thus time and energy consuming.

6.5. *Personal development and Skills Improvement*

During the elaboration of this work, I have gained new skills, and moreover, I have strengthened skills I already possess. The following list describes some important skills I consider I have gained after completing this work:

- Project management skills including time management, scope management and risk management,
- Requirements analysis skills,
- Virtualisation and the use of complex computing environments,
- FreeBSD and Open Source virtual routing,
- Fortification of my skills in Java programming and the use of design patterns,
- Python Programming skills,
- C++ advanced development skills,
- Academic research skills,
- Testing cryptographic program efficiency and simulating anonymity network functioning and features,

- Capacity to overcome problems that emerge during the different stages of the work.

Acknowledgements: I would like to thank my supervisor Professor Hassan KAZEMIAN for his guidance, support, and patience. Despite his busy schedules, Prof Hassan has been always available to guide me and share his knowledge and research expertise. I feel fortunate that I had him as supervisor. I would also like to thank my thesis reader and MSc project module leader Professor Karim OUZZANE for his time and valuable help. Finally, to MSc Computer Forensics and IT Security lecturers and especially Dr Deepthi N, RATNAYAKE who provided me with valuable advice and support. Dr Jiaming CAI and for all the wonderful discussions about security and privacy.

Frequently used terms and acronyms

Onion Proxy	OP is the client side of TOR (the software) running on behalf of the user machine and ensure communication with the network (create cells, perform encryption and decryption, manage circuit and routing ... etc.).
Onion Router	OR is TOR dedicated software router ensuring routing cells throughout the network and performing wrapping and un-wrapping (encryption/decryption).
Bandwidth	the volume of traffic (incoming and outgoing) that a OR could sustain. The information is retrieved from the operator claimed capacity or DA observed.
Directory Authority	DA is a special dedicated server managed by the TOR and maintains all the information about the ORs and links status.
Circuit	called also path is the route through the TOR network built by used by a client and consists of an entry (guard) OR, middle(s) OR and Exit OR.
Hidden Service	HS is location and functioning not public internet service that only TOR user can access.
AES	The Advanced Encryption Standard (Rijndael) is a secret key (symmetric) encryption.
RSA	public-key (asymmetric) encryption algorithm.
DH	Diffie–Hellman key exchange algorithm.
MAC	Message Authentication Code (also called MIC)
Hash	one way function used to map data of arbitrary size to data of fixed size for integrity and authenticity purposes.
CBC	the Cipher Block Chaining encryption mode.
CTR	Counter-mode encryption mode.
OCB	Offset Codebook Mode authenticated encryption mode.
CCM	Counter with CBC-MAC authenticated encryption mode.
GCM	Galois/Counter Mode authenticated encryption mode.
EAX	Authenticated Encryption with Associated Data (AEAD)
TCP/IP	Transmission Control Protocol/Internet Protocol.
TLS	Transport Layer Security.
Cell	the TOR equivalence for packets, a fixed size data structured into a specific way.

Appendix 1: Advanced Encryption Standard AES algorithm

AES is the most efficient symmetric encryption algorithm used in symmetric key cryptography as it depends on longer key sizes. The chosen algorithm behind the Advanced Encryption System label was the Rijndael algorithm. AES / Rijndael support different key lengths of 128, 192, and 256 bit key lengths. The longer the key length used the stronger and more difficult the encryption will be to break into. However using a 256 bit key to protect and encrypt data would also mean it will require more processing power and take longer to process.

Depending on the key lengths and block sizes AES produces a number of rounds of computation.

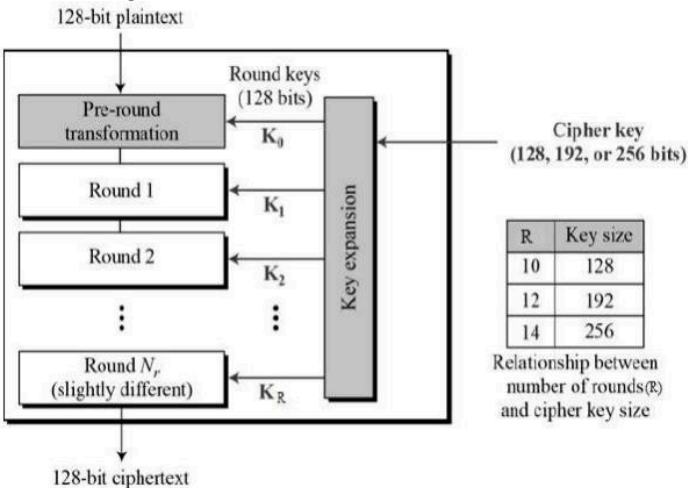
- In a block and key size of 128 bits, there are 10 computation rounds,
- In a block and key size of 192 bits, there are 12 computation rounds,
- In a block and key size of 256 bits, there are 14 computation rounds.

AES became the replacement for 3DES and DES. DES in particular was found to be weak and breakable. AES is a popular encryption standard approved by the government and supported by all security protocol and equipment's vendors (Smith, 2013). AES today is also used in removable media such as USB's and external hard drives. It is effective in both hardware and software and uses less memory than most other symmetric algorithms.

Simply put, you can protect data on encryption software running the AES algorithm. “If an encrypted USB was stolen and in the wrong hands, data would be protected and would be in an un-readable format” (Singh & Supriya, 2013) .

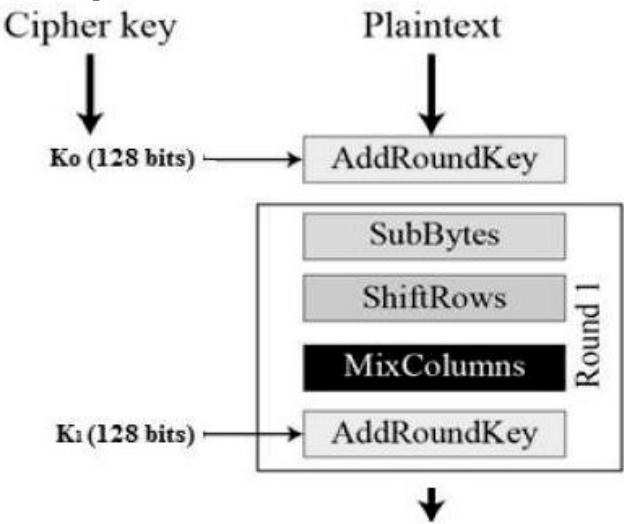
Operation of AES:

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations). Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix. Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key. The schematic of AES structure is given in the following illustration:



Encryption Process:

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below:



AES Functions:

Substitution (SubBytes): The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shiftrows: Each of the four rows of the matrix is shifted to the left. Any entries that ‘fall off’ are re-inserted on the right side of row. The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

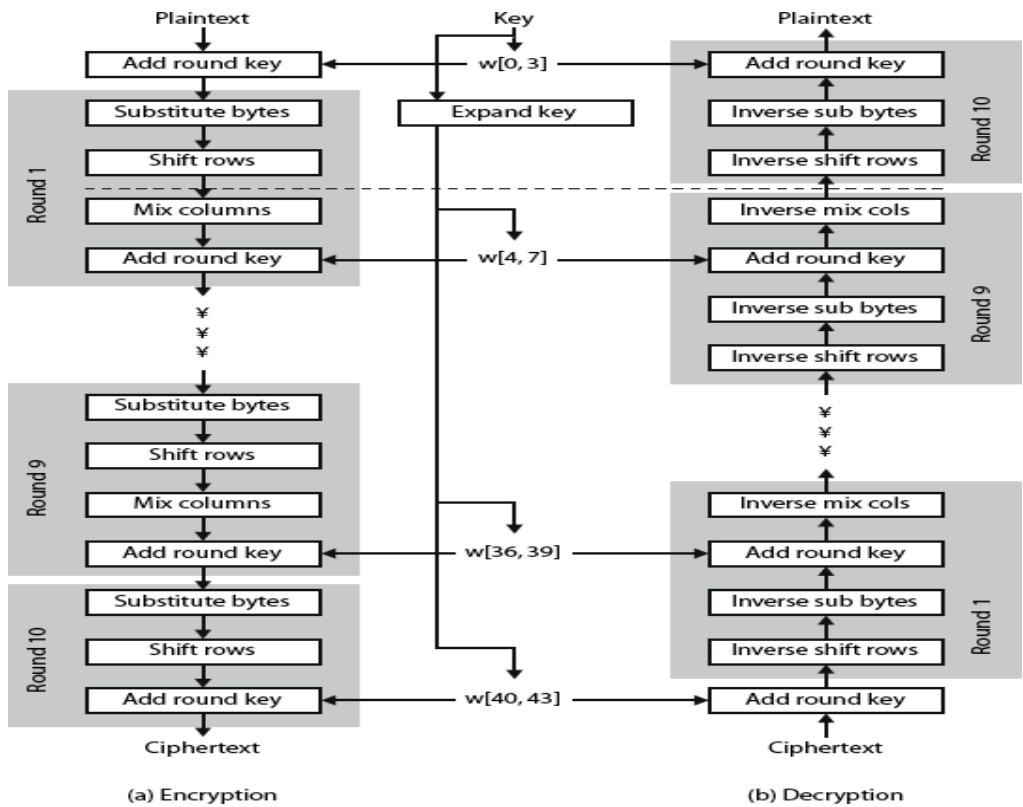
MixColumns: Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the

original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Addroundkey: The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

AES Analysis

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of ‘future- proofing’ against progress in the ability to perform exhaustive key searches. However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed



AES (Advanced Encryption Standard) detailed scheme

Appendix 2: Authenticated-encryption OCB Mode

OCB ENCRYPTION ALGORITHM

Input:

K, string of KEYLEN bits // Key
 N, string of no more than 120 bits // Nonce A,
 string of any length // Associated data P, string
 of any length // Plaintext

Output:

C, string of length $\text{bitlen}(P) + \text{TAGLEN}$ bits // Ciphertext C is
 defined as follows.

```
//
// Key-dependent variables
//
L_* = ENCIPHER(K, zeros(128))
L_$ = double(L_*) L_0 =
double(L_$)
L_i = double(L_{i-1}) for every integer i > 0
//
// Consider P as a sequence of 128-bit blocks
//
Let m be the largest integer so that  $128m \leq \text{bitlen}(P)$  Let P_1,
P_2, ..., P_m and P_* be strings so that
P = P_1 || P_2 || ... || P_m || P_*, and
 $\text{bitlen}(P_i) = 128$  for each  $1 \leq i \leq m$ . Note: P_*
may possibly be the empty string.
//
// Nonce-dependent and per-encryption variables
//
Nonce = num2str(TAGLEN mod 128,7) || zeros(120-bitlen(N)) || 1 || N bottom =
str2num(Nonce[123..128])
Ktop = ENCIPHER(K, Nonce[1..122] || zeros(6)) Stretch =
Ktop || (Ktop[1..64] xor Ktop[9..72]) Offset_0 =
Stretch[1+bottom..128+bottom] Checksum_0 = zeros(128)
//
// Process any whole blocks
```

```

//
for each 1 <= i <= m
Offset_i = Offset_{i-1} xor L_{ntz(i)}
C_i = Offset_i xor ENCIPHER(K, P_i xor Offset_i)
Checksum_i = Checksum_{i-1} xor P_i end for
//
// Process any final partial block and compute raw tag
//
if bitlen(P_*) > 0 then Offset_* =
Offset_m xor L_*
Pad = ENCIPHER(K, Offset_*) C_* = P_*
xor Pad[1..bitlen(P_*)]
Checksum_* = Checksum_m xor (P_* || 1 || zeros(127-bitlen(P_*)))
Tag = ENCIPHER(K, Checksum_* xor Offset_* xor L_$) xor HASH(K,A) else
C_* = <empty string>
Tag = ENCIPHER(K, Checksum_m xor Offset_m xor L_$) xor HASH(K,A) end if
//
// Assemble ciphertext
//
C = C_1 || C_2 || ... || C_m || C_* || Tag[1..TAGLEN]

```

OCB DECRYPTION ALGORITHM

Input:

K, string of KEYLEN bits // Key
N, string of no more than 120 bits // Nonce A,
string of any length // Associated data
C, string of at least TAGLEN bits // Ciphertext

Output:

P, string of length bitlen(C) - TAGLEN bits, // Plaintext or
INVALID indicating authentication failure
P is defined as follows.

```

//

```

```

// Key-dependent variables

//
L_$ = double(L_*) L_0 =
double(L_$)
L_i = double(L_{i-1}) for every integer i > 0

//
// Consider C as a sequence of 128-bit blocks

//
Let m be the largest integer so that 128m ≤ bitlen(C) - TAGLEN Let C_1,
C_2, ..., C_m, C_* and T be strings so that
C = C_1 || C_2 || ... || C_m || C_* || T,
bitlen(C_i) = 128 for each 1 ≤ i ≤ m, and
bitlen(T) = TAGLEN.
Note: C_* may possibly be the empty string.

//
// Nonce-dependent and per-decryption variables

//
Nonce = num2str(TAGLEN mod 128,7) || zeros(120-bitlen(N)) || 1 || N bottom =
str2num(Nonce[123..128])
Ktop = ENCIPHER(K, Nonce[1..122] || zeros(6)) Stretch =
Ktop || (Ktop[1..64] xor Ktop[9..72]) Offset_0 =
Stretch[1+bottom..128+bottom] Checksum_0 = zeros(128)

//
// Process any whole blocks

//
for each 1 ≤ i ≤ m
Offset_i = Offset_{i-1} xor L_{ntz(i)}
P_i = Offset_i xor DECIPHER(K, C_i xor Offset_i)
Checksum_i = Checksum_{i-1} xor P_i end for
//
// Process any final partial block and compute raw tag

//
if bitlen(C_*) > 0 then Offset_* =
Offset_m xor L_*
Pad = ENCIPHER(K, Offset_*)

```

```

P_* = C_* xor Pad[1..bitlen(C_*)]

Checksum_* = Checksum_m xor (P_* || 1 || zeros(127-bitlen(P_*)))

Tag = ENCIPHER(K, Checksum_* xor Offset_* xor L_*) xor HASH(K,A) else
P_* = <empty string>

Tag = ENCIPHER(K, Checksum_m xor Offset_m xor L_*) xor HASH(K,A) end if

//

// Check for validity and assemble plaintext

//

if (Tag[1..TAGLEN] == T) then P = P_1

|| P_2 || ... || P_m || P_* else

P = INVALID

end if

```

References

- AlSabah, M. and Goldberg, I. (2015). Performance and Security Improvements for Tor: A Survey. <http://eprint.iacr.org/2015/235>.
- AlSabah, M., Bauer, K., Elahi, T. and Goldberg, I. (2013a). The Path Less Travelled: Overcoming Tor's Bottlenecks with Traffic Splitting. In Privacy Enhancing Technologies - 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings. Springer, 143–163.
- AlSabah, M. and Goldberg, I. (2013b). PCTCP: Per-Circuit TCP-over-IPsec Transport for Anonymous Communication Overlay Networks. In Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, 349–360.
- AlSabah, M., Bauer, K., and Goldberg, I. (2012). Enhancing Tor's Performance Using Real-Time Traffic Classification. In Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS '12). ACM, New York, NY, USA, 73–84.
- AlSabah, M., Bauer, K., Goldberg, I., Grunwald, D., McCoy, D., Savage, S. and Voelker, G. (2011). DefenestraTor: Throwing Out Windows in Tor. In Privacy Enhancing Technologies. 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings. Springer Berlin Heidelberg, 134–154.
- Antonakakis, M., Edman, M. and Syverson, P. (2009). As-Awareness in TOR Path Selection. In Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, ACM 380–389.
- Backes, M., Goldberg, I., Kate, A., & Mohammadi, E. (2012). Provably Secure and Practical Onion Routing. *Computer Security Foundations Symposium (CSF)*, 25, 369-385.
- Bauer, K. and Sherr, M. (2011). ExperimentTor: A Testbed for Safe and Realistic Tor Experimentation. USENIX 2011, <http://www.usenix.org/events/cset11>.
- Bauer, K., McCoy, D., Sherr, M. and Grunwald, D. (2011). ExperimentTOR: A test-bed for safe and realistic tor experimentation. In Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test (CSET). Bogdanov, A., Lauridsen, M. and Tischhauser, E. (2014). AES-Based Authenticated Encryption Modes in Parallel High-Performance Software. IEEE library.
- Fu, X. and Ling, Z. (2009). One Cell is enough to break Tor's Anonymity. White Paper for Black Hat DC 2009.
- Boyd, W. (2011). A Simulation of Circuit Creation in Tor. Master thesis submitted at Wesleyan University, Connecticut April, 2011.
- Benmeziane, S., Badache, N. & Bensimessaud, S. (2011). Tor Network Limits. International Conference on Network Computing and Information Security, 1, 200-205.
- Burstein, A. J. (2008). Conducting cyber security research legally and ethically. *1st USENIX Workshop on Large-Scale Exploits and Emergent Threats*, Berkeley, CA, USA, pages 1-8. USENIX Association.

- Camenisch, J., Lysyanskaya, A. (2005). A formal treatment of onion routing. 25th Annual International Conference in Advances in Cryptology CRYPTO 2005, 169-187.
- Carnielli, A. and Aiash, M. (2015). Will TOR Achieve its Goals in the Future Internet? An Empirical Study of using TOR with Cloud Computing. 2015 29th International Conference on Advanced Information Networking and Applications Workshops.
- Casenove, M., Miraglia, A. (2014). Botnet over Tor: The Illusion of Hiding. 6th International Conference on Cyber Conflict P.Brangetto, M.Maybaum, J.Stinissen (Eds.), NATO CCD COE Publications, Tallinn.
- Castelluccia, C., De Cristofaro, E. and Perito, D. (2010). Private information disclosure from web searches. In Mikhail J. Atallah and Nicholas J. Hopper, editors, *Privacy Enhancing Technologies*, 6205 of Lecture Notes in Computer Science, 38-55.
- Dahal, S., Lee, J., Kang, J. and Shin, S. (2015). Analysis on End-to-End Node Selection Probability in TOR Networking, IEEE ICOIN 2015 ISBN: 978-1-4799-8342-1/15.
- Danezis, G. Diaz, C. and Syverson, P. (2010). Systems for Anonymous Communication. In *CRC Handbook of Financial Cryptography and Security, CRC Cryptography and Network Security Series*, B. Rosenberg, and D. Stinson (Eds.), 341-390.
- Darcie, W., Boggs, R., Sammons, J. and Fenger, T. (2013). Online Anonymity: Forensic Analysis of the Tor Browser Bundle. ICDPSC 2013.
- Dingledine, R. and Mathewson, N. (2016a). TOR Directory Specification. <https://gitweb.etorproject.org/torspec.git/tree/dir-spec.txt>. (2016). Accessed March 2016.
- Dingledine, R. and Mathewson, N. (2016b). TOR Protocol Specification. <https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt>. (2016). Accessed March 2016.
- Dingledine, R., Mathewson, N. & Syverson, P. (2004). Tor: The Second-Generation Onion Router. 13th Security Symposium (USENIX), 303-320.
- Dingledine, R., Mathewson, N., Murdoch, S. & Syverson, P. (2014). Tor: The Second-Generation Onion Router Draft 2014. <http://www.cl.cam.ac.uk/>, Accessed on 11-02-2014.
- Douceur, J. (2002). The Sybil Attack. In: Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002). Volume 2429 of LNCS, Springer.
- Feigenbaum, J., Johnson, A. and Syverson, P. F. (2007). Probabilistic analysis of onion routing in a black-box model. 6th ACM Workshop on Privacy in the Electronic Society (WPES), 1-10.
- Goldberg, I., Stebila, D. and Ustaoglu, B. (2012). Anonymity and one-way authentication in key exchange protocols. IEEE ICPC 2012.
- Haraty, R.A. & Zantout, B. (2014). The TOR Data Communication System: A Survey. *Journal of Communications and Networks*, 16, 415-420.
- Huhta, O. (2014). Linking Tor Circuits. MSc Information Security dissertation submitted to University College London.
- Jansen, R., Geddes, J., Wacek, C., Sherr, M. and Syverson, P. (2014). US Never Been KIST: Tor's Congestion Management Blossoms with Kernel-Informed Socket Transport. *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA* ISBN 978-1-931971-15-7.
- Johnson, A., Wacek, C., Jansen, R., Sherr, M. and Syverson, P. (2010). Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries, *Association for Computing Machinery*, ACM, US.
- Kate, A. and Goldberg, I. (2010). Distributed Private-Key Generators for Identity-Based Cryptography. 7th Conference on Security and Cryptography for Networks (SCN), 436-453.
- Krovetz, T. and Rogaway, P. (2014). OCB implementation and performance analysis. IETF RFC publications.
- Lazzari, M. (2014). Systematic Testing of Tor. Submitted as Master Thesis, ETH Zurich.
- Ling, Z., Luo, J., Yu, W. and Fu, X. (2011). Equal-sized Cells Mean Equal-sized Packets in TOR. IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings.
- Marquis, M. (2013). For their eyes only. The Commercialization of Digital Spying citizen lab Canada global security research.
- Mccoy, D., Bauer, K., Grunwald, D., Kohno, T. and Sicker, D. (2008). Shining light in dark places: Understanding the tor network. 8th international symposium on Privacy Enhancing Technologies, PETS '08, 63-76, Berlin.
- Murdoch, S. and Watson, R. (2007). Metrics for Security and Performance in Low-Latency Anonymity Systems. University of Cambridge, UK.
- Nia, M.A., Karbasi, A.H. & Atani, R.E. (2014). Stop Tracking Me: An anti-detection type solution for anonymous data. 4th International eConference on Computer and Knowledge Engineering (ICCCKE), 14, 685-690.
- Øverlier, L. and Syverson, P. (2006). Locating hidden servers. In Proceedings of the 2006 IEEE Symposium on Security and Privacy, Oakland, CA, US, IEEE Computer Society.
- Perry, M. (2007). Securing the Tor Network, Black Hat USA 2007 Supplementary Handout.
- Reardon, J. and Goldberg, I. (2010). Improving TOR using a TCP-over-DTLS Tunnel. TOR project research papers, <https://gitweb.etorproject.org>.
- Schanck, J., Whyte, W. and Zhang, Z. (2015). A quantum-safe circuit-extension handshake for Tor. Security innovation white paper.

- Singh, S. (2015). Large-Scale Emulation of Anonymous Communication Networks. Matser thesis presented to the University of Waterloo.
- Snader, R. and Borisov, N. (2008). A tune-up for Tor: Improving security and performance in the Tor network. Network & Distributed System Security Symposium, Internet Society.
- Soghoian, C. (2011). Enforced Community Standards for Research on Users of the Tor Anonymity Network. *Second Workshop on Ethics in Computer Security Research WECSR*, 02, St. Lucia.
- Stupples, D. (2013). Security Challenge of TOR and the Deep Web. The 8th International Conference for Internet Technology and Secured Transactions ICITST 2013.
- Svenda, P. (2012). Basic comparison of Modes for Authenticated-Encryption (IAPM, XCBC, OCB, CCM, EAX, CWC, GCM, PCFB, CS). Masaryk University in Brno.
- Syverson, S., Goldschlog, D. and Reeds, M. (1997). Anonymous connections and onion routing. *Proceedings of the IEEE Symposium on Security and Privacy*, USA, 482-494.
- TOR Deployment. (2016). TOR network detailed deployment. <https://abouttor.tor.org>, Accessed on April 2016.
- TOR Flow. (2016). TOR flux across the world, <https://torflow.uncharted.software> 2016-1-13, accessed on April 2016.
- TOR Project. (2016). TOR active users number in UK. <https://metrics.torproject.org>, accessed on April 2016. TOR Metrics. (2016). TOR Network overall bandwidth, <https://metrics.torproject.org/bandwidth.html>, accessed on April 2016.
- Wacek, C., Tan, H., Bauer, K. and Sherr, M. (2013). An Empirical Evaluation of Relay Selection in TOR. In *Proceedings of the Network and Distributed System Security Symposium - NDSS'13*, The Internet Society.
- Yenuguvanilanka, J. and Elkeelany, O. (2007). Performance Evaluation of Hardware Models of Advanced Encryption Standard (AES) Algorithm. Tennessee Tech University.
- Zhang, Y. (2009). Effective attacks in the tor authentication protocol. *3th International Conference on Network and System Security*, 09, 81-86.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.