Anonymity Network: Cloud based onion routing

Fabian Grünbichler Klaus Krapfenbauer Thomas Hackl
Stefan Gamerith Milan Dzenovljanovic

ABSTRACT

An anonymity network should enable users to access the public Web while, at the same time, hiding their identity on the Internet. Onion routing's anonymous connections are strongly resistant to traffic analysis and can be used through a socket connection. In an onion network, messages are encrypted and transmitted over a onion routers. Every onion router removes the layer of encryption, reads the routing instructions and sends the message to the next router in the chain where the same process is repeated. In the onion network chain every router is aware just about existence of the previous end next node, and only the last node knows the content of a data. Todays most used global onion routing network is Tor. Because of using a large number of volunteers, Tor faces a poor performance often. Therefore as well as for benefits of Cloud computing (broad network access, resource pooling, rapid elasticity, ensured services), moving onion routing services into the "cloud" seems like a logical step. This paper describes anonymous connections and their implementation using Cloud-based onion routing.

General Terms

Security, Privacy

Keywords

Anonymity networks, Onion routing, Cloud computing

1. INTRODUCTION

Anonymity networks enable users to communicate over the public Web while hiding their identity from one another. Nevertheless, is Internet communication private? Greatest concerns today is preventing outsiders listening in on electronic conversation. But *traffic analysis* can also leak some sensitive information because encrypted messages can be tracked, revealing who is talking to whom. [anonymuos connections and onion routing introduction]

Todays best and most popular tool for accessing the Internet anonymously is Tor. By taking the traffic through a sequence of nodes, Tor allows the traffic to appear to target from its last node, not from the user directly. Main disadvantage of Tor is that its relays are hosted by a volunteers all over the world. This volunteers mainly use private, consumer-grade ISP connections which often have limited bandwidth capacity, therefore Tor faces performance drops. [a proposal for COR]

Moving onion routing services to the "cloud" will benefit with possibility to rotate between numbers of IP addresses as well as with large number of high-quality, high-bandwidth nodes. Great advantage of the "cloud" infrastructure lays in its "elasticity". By spawning new virtual machine instance in the "cloud" one can easily add or remove a new node to a "cloud". [a proposal for COR]

Main goal of *traffic analysis* is to discover participants of conversation, weather the participants are exchanging mails or user is just browsing the Web. Bearing this in mind, the *anonymity networks* are developed to obfuscate the observers by making them difficult to identify informations, that are stored into the packet headers or in a track of encrypted payload, from the connection. Thus, any information that carries identifying data must be sent through the *anonymus network*. Onion routing is implementation of *anonymous network* that stands out with low latency and overhead. Fundamentally, onion routing relays on Tor's protocol and provides two-way, real-time, connection-based communication that does not require the target to participate in the *anonymity* protocol. All afore mentioned features together make anonymising communication on the Web easier.

2. ONION ROUTING

2.1 Overview

In onion routing, user connects to a target over a sequence of machines called *onion routers*, rather than connect directly to a target. This way the *originator* and the target remain anonymous to *observer*. It is possible for *originator* to remain anonymous to the target. In this case all informations that identifies the *originator* have to be removed from the data stream before being sent over the *onion routing network*.

2.2 Data transfer

In the *onion routing network* routers communicates over a TCP (it can be implemented on top of other protocols). The

set of the *onion routers* in a route is defined at very setup of the connection, when *originator* creates an *onion*, and each router knows only previous and next router along the route. An *onion* is a layered data structure that encapsulates cryptographic informations such as the keys and cryptographic algorithms used during the data transfer. Along the route each *onion router* uses its public key do decript the entire *onion*. This way the identity of the next router along the route as well as embedded *onion* is exposed. The *onion router* ends the data to the next router and after the connection is established, the data can be sent in both directions. Using the keys and algorithms from the onion, the data is encrypted before it is sent from the originator. During the data transfer along the route, each *onion router* removes one layer of encryption, like peeling an onion, so at the target the data arrives as a plain text. Because of the layered public key cryptography an *onion* looks different to every *onion router* along the connection. [onion routing for anonymous and private internet connection]

2.3 Overhead

Because of the public key decryption, the most expensive calculation in the onion routing network is public-key cryptography, and it only takes place at the connection setup. In the data movement phase only the symmetric-key cryptography is used, which is much faster and can be as fast as ordinary link encryption. Delay in the data transfer depends on the number of the *onion routers*. Overall, overhead in the *onion routing* is sufficiently small. [Onion Routing for Anonymus and Private Internet Connection]

3. FORMAL DESCRIPTION

Let say we have N number of routers, where each router have its own public (Su) and private (Sr) key. Public key is known to the *originator* and private keys are known to the router. [Onion Routing for Anonymous Communications]

As outlined before, *onion routing protocol* relays on the public-key cryptography, therefore functions for encryption *E[key](data)* and decryption *D[key](data)* are needed. Key par in the public-key cryptography is generated in the way that data encrypted with public key Su can be decrypted only with private key Sr, and vice versa, i.e. *D[Su](E[Sr](data)) = data* and *D[Sr](E[Su](data)) = data*. [Onion Routing for Anonymous Communications]

When *originator* wants to send informations over the *onion routing network*, he sends a request to the *onion proxy* and gets back the set of randomly chosen *onion routers*, e.g. <4, 3, 5>. This means that information, which travels from the *originator* to the *target*, will be routed over this three routers respectively. The first router in the circuit[1] is called *entry node* and the last node is called *exit node*. Before information is sent to the *entry node*, *onion proxy* creates an *onion*, which in our case looks like: *E[4u](3's IP address, E[3u](5' s IP address, E[5u](data)))*. The *onion* is then forwarded to the *entry node* (4). The *entry node* decrypts the *onion* with its private key, reads the IP address of the next node (3) and passes the the data to it where the same process is repeated. [Onion Routing for Anonymous Communications]

[1]Circuit represents the logical connection between two nodes

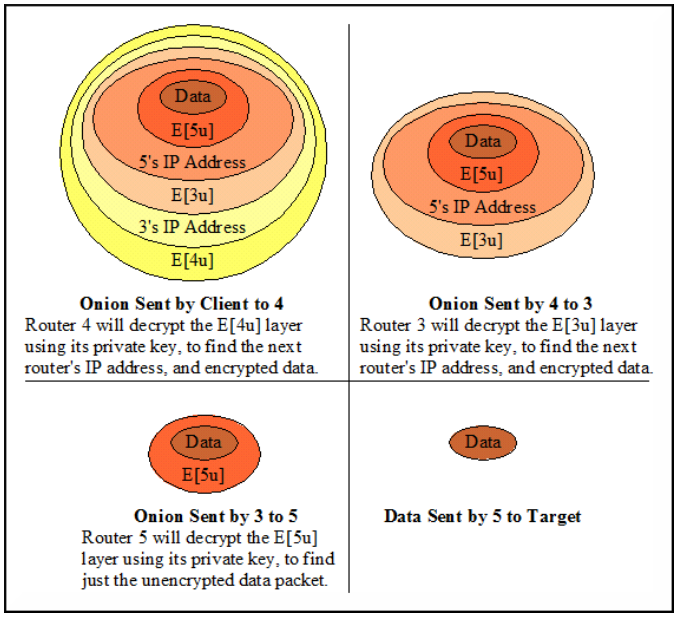

Figure 1: Onion routing [citation]

It is possible that each router has a number of TCP connection to the other routers, and multiple circuits between two same nodes might use the same TCP connection.Therefore, every router needs to remember the association between two circuits. [Onion Routing for Anonymous Communications]

When the circuit <4, 3, 5> is established it allows data to travel in both ways without including routing information. In the example above, when a response message from the *target* arrives at the *exit node* (5), (5) already know to which circuit and node (3) it should forward the information. (5) first encrypt the data it received from the *target* with its private key and sends it to (3). (3) encrypts the data with its private key and sends it to (4). (4) does the same thing and passes the data to the *onion proxy*. In order to read the original information, the *onion proxy* decrypts the data it received from (4) as: *D[4u](D[3u](D[5u](encrypted onion)))*. [Onion Routing for Anonymous Communications]

3.1 Type Changes and *Special* Characters

We have already seen several typeface changes in this sample. You can indicate italicized words or phrases in your text with the command `\textit`; emboldening with the command `\textbf` and typewriter-style (for instance, for computer code) with `\texttt`. But remember, you do not have to indicate typestyle changes when such changes are part of the *structural* elements of your article; for instance, the heading of this subsection will be in a sans serif[2] typeface, but that is handled by the document class file. Take care with the use of[3] the curly braces in typeface changes; they mark the beginning and end of the text that is to be in the different typeface.

[2]A third footnote, here. Let's make this a rather short one to see how it looks.
[3]A fourth, and last, footnote.

You can use whatever symbols, accented characters, or non-English characters you need anywhere in your document; you can find a complete list of what is available in the *LaTeX User's Guide*[**?**].

3.2 Math Equations

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

3.2.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin. . .\end` construction or with the short form `$. . .$`. You can use any of the symbols and structures, from α to ω, available in LaTeX[**?**]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n\to\infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

3.2.2 Display Equations

A numbered display equation – one set off by vertical space from the text and centered horizontally – is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\to\infty} x = 0 \tag{1}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_{0}^{\pi+2} f \tag{2}$$

just to demonstrate LaTeX's able handling of numbering.

3.3 Citations

Citations to articles [**?**, **?**, **?**, **?**], conference proceedings [**?**] or books [**?**, **?**] listed in the Bibliography section of your article will occur throughout the text of your article. You should use BibTeX to automatically produce this bibliography; you simply need to insert one of several citation commands with a key of the item cited in the proper location in the `.tex` file [**?**]. The key is a short reference you invent to uniquely identify each work; in this sample document, the key is the first author's surname and a word from the title. This identifying key is included with each item in the `.bib` file for your article.

The details of the construction of the `.bib` file are beyond the scope of this sample document, but more information can be found in the *Author's Guide*, and exhaustive details in the *LaTeX User's Guide*[**?**].

This article shows only the plainest form of the citation command, using `\cite`. This is what is stipulated in the SIGS style specifications. No other citation format is endorsed.

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
\$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

3.4 Tables

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper "floating" placement of tables, use the environment **table** to enclose the table's contents and the table caption. The contents of the table itself must go in the **tabular** environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on **tabular** material is found in the *LaTeX User's Guide*.

Immediately following this sentence is the point at which Table 1 is included in the input file; compare the placement of the table here with the table in the printed dvi output of this document.

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 2 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed dvi output of this document.

3.5 Figures

Like tables, figures cannot be split across pages; the best placement for them is typically the top or the bottom of the page nearest their initial cite. To ensure this proper "floating" placement of figures, use the environment **figure** to enclose the figure and its caption.

This sample document contains examples of **.eps** and **.ps** files to be displayable with LaTeX. More details on each of these is found in the *Author's Guide*.

As was the case with tables, you may want a figure that spans two columns. To do this, and still to ensure proper "floating" placement of tables, use the environment **figure*** to enclose the figure and its caption.

Note that either **.ps** or **.eps** formats are used; use the `\epsfig` or `\psfig` commands as appropriate for the different file types.

Table 2: Some Typical Commands

Command	A Number	Comments
`\alignauthor`	100	Author alignment
`\numberofauthors`	200	Author enumeration
`\table`	300	For tables
`\table*`	400	For wider tables

Figure 2: A sample black and white graphic (.eps format).

Figure 3: A sample black and white graphic (.eps format) that has been resized with the `epsfig` command.

3.6 Theorem-like Constructs

Other common constructs that may occur in your article are the forms for logical constructs like theorems, axioms, corollaries and proofs. There are two forms, one produced by the command `\newtheorem` and the other by the command `\newdef`; perhaps the clearest and easiest way to distinguish them is to compare the two in the output of this sample document:

This uses the **theorem** environment, created by the `\newtheorem` command:

THEOREM 1. *Let f be continuous on $[a, b]$. If G is an antiderivative for f on $[a, b]$, then*

$$\int_a^b f(t)dt = G(b) - G(a).$$

The other uses the **definition** environment, created by the `\newdef` command:

Definition 1. If z is irrational, then by e^z we mean the unique number which has logarithm z:

$$\log e^z = z$$

Two lists of constructs that use one of these forms is given in the *Author's Guidelines*.

Figure 4: A sample black and white graphic (.ps format) that has been resized with the `psfig` command.

and don't forget to end the environment with figure*, not figure!

There is one other similar construct environment, which is already set up for you; i.e. you must *not* use a `\newdef` command to create it: the **proof** environment. Here is a example of its use:

PROOF. Suppose on the contrary there exists a real number L such that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = L.$$

Then

$$l = \lim_{x\to c} f(x) = \lim_{x\to c} \left[gx \cdot \frac{f(x)}{g(x)}\right] = \lim_{x\to c} g(x) \cdot \lim_{x\to c} \frac{f(x)}{g(x)} = 0 \cdot L = 0,$$

which contradicts our assumption that $l \neq 0$. □

Complete rules about using these environments and using the two different creation commands are in the *Author's Guide*; please consult it for more detailed instructions. If you need to use another construct, not listed therein, which you want to have the same formatting as the Theorem or the Definition[**?**] shown above, use the `\newtheorem` or the `\newdef` command, respectively, to create it.

A *Caveat* for the TEX Expert

Because you have just been given permission to use the `\newdef` command to create a new form, you might think you can use TEX's `\def` to create a new command: *Please refrain from doing this!* Remember that your LATEX source code is primarily intended to create camera-ready copy, but may be converted to other forms – e.g. HTML. If you inadvertently omit some or all of the `\def`s recompilation will be, to say the least, problematic.

4. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LATEX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

5. ACKNOWLEDGMENTS

Figure 5: A sample black and white graphic (.eps format) that needs to span two columns of text.

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

APPENDIX

A. HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used to indicate the start of each Appendix, with alphabetic order designation (i.e. the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

A.1 Introduction

A.2 The Body of the Paper

A.2.1 Type Changes and Special Characters

A.2.2 Math Equations

Inline (In-text) Equations

Display Equations

A.2.3 Citations

A.2.4 Tables

A.2.5 Figures

A.2.6 Theorem-like Constructs

A Caveat for the TEX Expert

A.3 Conclusions

A.4 Acknowledgments

A.5 Additional Authors

This section is inserted by LaTeX; you do not insert it. You just add the names and information in the `\additionalauthors` command at the start of the document.

A.6 References

Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the .bbl file. Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.

B. MORE HELP FOR THE HARDY

The acm_proc_article-sp document class file itself is chockfull of succinct and helpful comments. If you consider yourself a moderately experienced to expert user of LaTeX, you may find reading it useful but please remember not to change it.