# Hydra – Anonymous Network Routing Mechanism

**Mrinal Wahal[1], Tanupriya Choudhury[2]**

*[1,2]Amity University Uttar Pradesh*
*[1] MrinalWahal@gmail.com, [2] tchoudhury@amity.edu*
*Hydra – Anonymous Network Routing Mechanism*

***Abstract: Anonymous network routing mechanisms are an epitome of network security to securely transmit data over a network. Example: Onion routing, Garlic routing. It's afoolproof mechanism to route data packets across the network with supposed source and destination anonymity, along with end-to-end encryption of data and provide protection from eave-droppers, surveilling government agencies, hackers and similar unwanted agencies. Except that these existing mechanisms are NOT completely fool-proof. And this paper will explain how along with introducing a considerably better mechanism.***

***Keywords: Client-server systems, Computer hacking, Computer networks, Cryptographic protocols, Intelligent networks, Network security, Next generation networking, Routing protocols.***

## I. INTRODUCTION

This paper highlights the drawbacks of existing anonymous routing mechanisms and proposes a new mechanism, Hydra, to deploy a secure decentralized network for encrypted communications. Moreover, considering the fact that the [2] right to remain anonymous is a fundamental right crucial to maintain a healthy democracy, a new model like Hydra was strongly solicited in the modern times.Basic analogy of transferring data packets securely is to encrypt them using the infamous public key-private key cryptography method. Hydra also utilizes the same method, rather with a few modifications.

The rest of the portion of this paper is prepared which are as follows:

- Position of art work.
- Basic blueprint followed by all routing mechanisms.
- Drawbacks and disadvantages of existing concepts.
- Cyber-attacks possible on existing works.
- Model of operation which Hydra works on.
- End-to-end encryption utilized by Hydra.
- Advantages and Back falls of implementing Hydra.

## II. STATE OF ABILITY OF WORK

Since ages the question of individual privacy has bothered fundamentalists and judiciaries worldwide. How to decide what gives the governments worldwide a declared right to monitor its citizens? Example: US Domestic Surveillance Program. It has always been a tough fight between individual privacy and interpretation of laws of the Constitution. Therefore, quite a few concepts/mechanisms are being invented since decades to provide utmost privacy to individual users against any unethical monitoring agencies.

## III. RELATED WORKS

Among the very firsts of such anonymous routing mechanisms designed was the one proposed and doctored by D. Chaum, who invented the infamous mix nets which allowed to transfer data with layered encryptions over long distances. Eventually this technology came to be employed by various huge organizations/companies to implement a layer of security over their internal networks. This technology alsoultimately gave birth to the Onion Routers, wherein data is encrypted in layers from individual private keys of every single node present and participating in the network, like an onion. It, at the most fundamental level, employs free-routing and an additional decryption chain. Then came the TOR: Second generation onion router which infamously provided a complete graphical user interface to the non-geeky audience in order to provide them with a ready-made gateway to access the dark web without any prior technical knowledge which exempted them of the problem of manually installing and implementing tor on their existing nodes/systems. However, this so called secure and flawless routing technique was susceptible to traffic analysis attacks, majorly. Following this invention, soon arrived new concepts from great minds worldwide, like the ones described ahead in the paper. One of the very recent small-scale inventions include a new protocol by Ariadne [11] to withstand node compromise and rely on highly efficient symmetric cryptography in order to maximize security and minimize compromise. Everyone tried to maximize security and anonymity and therefore, went ahead on a quest to devise a way for both of them to go hand in hand as far as possible. One such technology was the Klein Bottle routing which proposed a distributed encryption over the original semantic encryption [9]. This particular routing technique at the most basic level proposes to fill the gap between onion routing and mix networks. However, like every mentioned technique even Klein Bottle routing failed to properly hide few drawbacks. One such backfall of this particular technique is that this technique employs distributed decryption to original encryption, therefore, knowledge of some partial private keys

978-1-5386-0514-1/17/$31.00 ©2017 IEEE

must be learnt when analyzing semantic security. Another quite recent invention of variance in onion routing mechanism was introduced by Kazuya Sakai, Min-Te Sun, Wei-Shinn Ku and Jie Wuwhich proposed a framework capable of outperforming onion-based, anonymous Epidemic, and zone-based routing.However, every other technique/technology had some major drawbacks. Though, one of the most advanced and nearly flawless technique/mechanism I encountered while researching for this project were Cashmere Routing, presented by Li Zhuang, Ben Y. Zhao, Antony Rowstron, and Riffle Networks.

Designers of the strong enough Riffle Networking mechanism defined the project as a considerable strong and efficient communication system with strong anonymity. The concept of attaching a smart mathematically viable proof in order to elevate the security and downgrade subjection to corruption of data by intermediate unethical hacking/surveillance agencies like hackers, government bodies or college/school administration, was employed in Hydra from the Riffle mechanism only.

## IV. COMMON BLUEPRINT OF OPERATION

Every routing mechanism generally follows the concept of transferring data in form of discrete packets over every single node existing in a network which has to be recurred by employing an anonymity technique/mechanism. And this data is encrypted with layers of public key encryption by each and every node as the data passes through every them. Eventually the data is unwrapped (decrypted) with private keys of every specific node in correspondence with their public key employed in the encryption process, one layer at a time, like an onion. [4] This protects the data with utmost level of security from intermediate unethical agencies from hijacks of any kind.

## V. DRAWBACKS OF EXISTING TECHNOLOGIES

1. Chaum Mixes & The Onion Routing: Chaum proposed wrapping the data with a new layer of encryption everytime it passes through a node, in the mix net. [5] However, if one node in the process breaks/is compromised, the entire routing path is dissolved and a new path has to be set up from scratch. In addition to that, it's time consuming, unsafe, consumes huge amount of band-with, displays a low-latency design [3], delays to recover all backtracked private keys used to encrypt data at every node and ultimately, it's computationally expensive.

2. Cashmere Routing[8]: This mechanism uses relay groups instead of single nodes for data transmission. Considering unsolicited data is received on the last relay group, unnecessary loss of band-with and memory takes place.

3. Mixmaster, Mix minion, Babel: Such technologies tried to maximize anonymity at the cost of latency. Therefore, deemed unsuitable/unfit for various applications such as web browsing. [1]

4. Pipenets: Again, a low latency design but allows the user to shut down the entire network without sending.

5. Crowds [14]: Aims to provide anonymity for web transactions by using layered encryption but without any public keys, therefore any node on the path can read the user's traffic.

6. Tarzan [15]: Also uses layered encryption and multi-hop routing. Meaning, it adds a new layer of encryption over the data with a new pair of keys. Thereby causing computational overhead and delay. Moreover, any node failure causes the entire path to be dissolved and be rebuilt which again is time consuming.

7. Aqua: Claims to provide traffic analysis resistance but it's not professionally possible to determine which mixnet is actually uncompromised. And assuming powerful adversaries are controlling both ends of the circuit, they can very readily de-anonymize clients. [13]

8. Negative social image of existing mechanisms: Existing technologies like the onion routing and garlic routing have lately developed a negative image due to their user base which ultimately reflects the decreased employability of such mechanisms in professional environments.

## VI. POSSIBLE ATTACKS WITH EXISTING TECHNIQUES

### *1. Major Passive Attacks [6]*

a. Observing traffic patterns: While the data at the source and destination ends is completely encrypted, the traffic between the nodes may not be encrypted. And therefore, the traffic patterns are subject to scrutiny, which may further help the attacker/hacker to pacify future attacks.

b. Website fingerprinting: An attacker may very easily build a database of file sizes and access patterns for certain websites. And may later cross-check the user's connections with the targeted website by referring to the database.

### *2. Major Active Attacks [7]*

a. Unethical Node: Considering the existing models work on single node relaying, then if any intermediate node colludes with the attacker to share the data or private keys, then the attacker may very well permanently damage the entire path and ultimately the data each and every time it passes through that path.

b. Re-encryption of data: Though, the actual data may not be directly visible to the attacker, however, the encrypted

data can still be sniffed and therefore, it allows the attacker to add an additional layer of encryption using his/her own keys, thereby, permanently damaging the data. Since, the data will not be decrypted properly by the destination.

c. Flooding Introduction Points: Assuming the attacker is well versed with causing buffer overflow attacks to carry out denial of service, the attacker can therefore, flood the introduction points on every node by generating too many requests to eventually overflow the receiving capacity of the node, thereby shutting it down.

d. Hostile Code at Exit Points: If the attacker adds an additional hostile code of any damaging capacity to the data at the exit node, then it practically destroys the efforts put in by the entire mechanism in the first place. Considering, it will not only need to re-encrypt the data in the first place, but also restart a new relaying path. This eventually results in delayed delivery time also.

## VII. HYDRA'S MODEL OF OPERATION/ALGORITHM

### *Initial Setting*

Let's assume me, Mrinal Wahal (source) and you (destination) wish to communicate and transfer confidential information without compromising source/destination anonymity. For the model to operate at best case scenario, we consider physical access to machines extremely important to monitor our data, but for worst case scenario we shall assume there's some unethical agency (individual hacker, government body, college surveillance team) already monitoring our traffic and therefore our communication is subject to scrutiny. The following figure 1 shows just another normal network without any pre-installed Hydra.

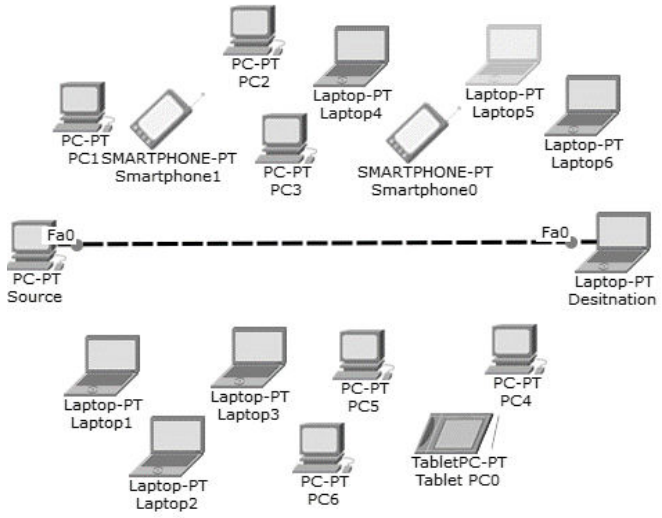


**Fig. 1. Normal network without installing Hydra.**

*Step 1:* The above network with 17 total nodes (including source and destination) was used for testing purposes along with an unethical agency is monitoring the traffic between us. Therefore, it wouldn't be categorically tough for them to get hold of the information we are transferring. Even if we are using secured protocols like HTTPS or even Secure Shells or any service with a weak 128-bit SSL encryption, which is fairly easy to crack with modern technologies.

Hydra works by dividing the existing and useable nodes on the network into **relay groups.** For gamma testing purposes, the source was freely allowed to choose the path to the destination as per the preferred definition by specifying the intermediate nodes/relay groups on the network which would pass the data further. For final design, these intermediate relay groups are randomly selected using automatic algorithms. And probabilistic algorithms were employed to deduce the fastest yet the least predictable source-destination relay path with maximum affordable degree of randomness. There's no viable way to analyze/judge the pattern employed by such algorithms to predict the relay path. The following figure 2 shows the next step - to divide the existing nodes of the test network into relay groups containing equal number of nodes among them. The destination is also included as a relay member in the last relay group on the path, as per Hydra's blueprint.

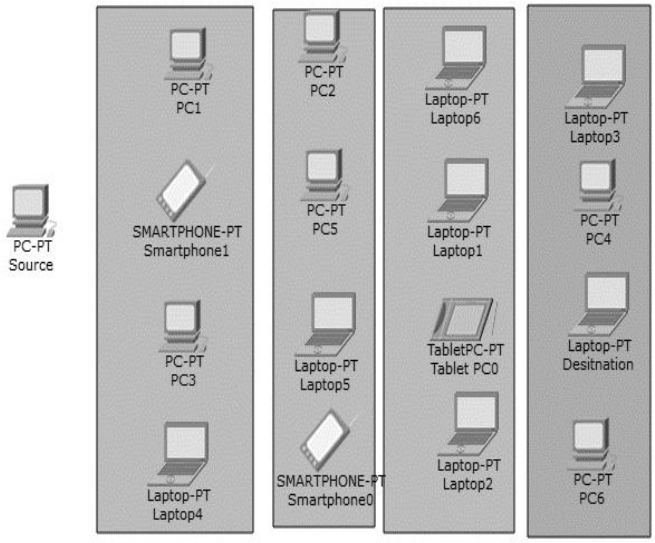


**Fig. 2. Dividing useable nodes into Relay Groups.**

*Step 2:* Hydra will automatically send the data to all the nodes of the first relay group. These relay groups will be generated on totally random bases. There's no viable way to predict the possible combination of these nodes just like the incapability of any computational agency to predict relay paths used to transport the data. The following figure3 gives a pictorial representation of transmitting the initial data from the source and transporting it to all the live nodes on the first relay group in the process.

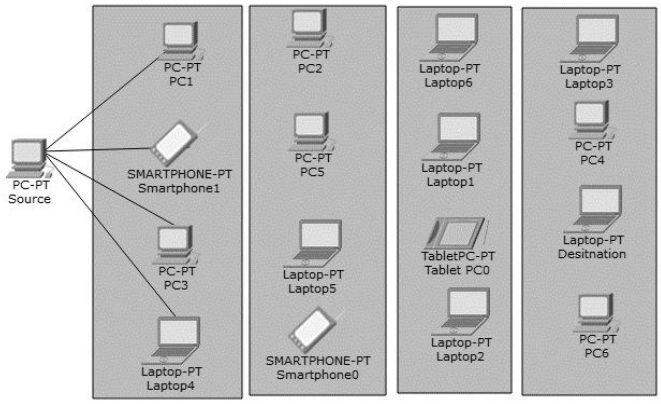


**Fig. 3. Broadcasting data to first relay group.**

*Step 3:* Similarly, as before, one node out of the current relay group will relay/transmit the data to all nodes of the next relaygroup until it reaches the destination, which will preferably be included in the last relay group. And can be randomized before the last group, when the scale of project reaches large number of nodes or in other words, during the phase of beta-testing. This relay path was kept user defined for gamma-testing purposes.The following figure number 4 shows extended broadcasting of encrypted data further to future nodes of the proceeding relay groups. This process is followed until the data finally reaches the destination in the very last relay group. It is also to be duly noted that only one node is required to send the data further from/to each relay group and in case of failure from that one node, other nodes are signaled/informed to transmit the data.

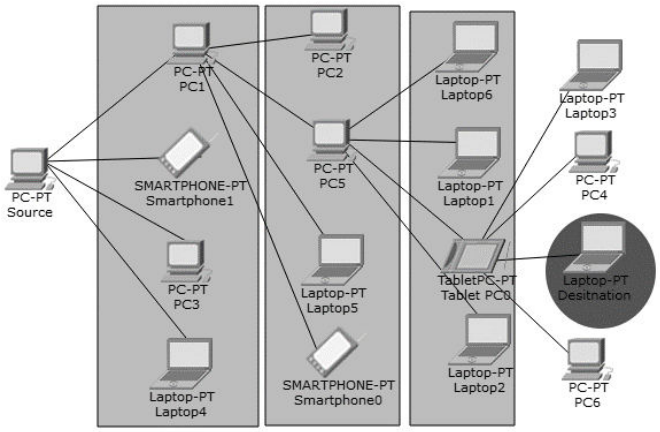


**Fig. 4. Further broadcasting data to next relay group until it reaches the destination.**

## VIII. END TO END ENCRYPTION

Hydra, too, operates on the infamous Public Key-Private Key encryption method. Meaning, the data will be encrypted using the destination's public key and therefore, will only be decrypted using the destination's private key, thereby, eliminating hampering or visibility of the transmitting data to any eavesdropping agency involved. The keys will be generated through the Diffie-Hellman Key Exchange mechanism and algorithm employed for encryption would be AES algorithm.

What if the hacker captures the data, adds one more layer of encryption with his/her own key and thereby, corrupts the data during transmission?

Hydra, works to resolve this issue by attaching another randomly generated mathematical proof along with the original data.This mathematical evidence shall be confirmed by every node to confirm that the transmitted data hasn't been corrupted in any possible manner by the previous node.Therefore, the final data to be transferred eventually becomes:

**d = e + p**

Where, d = Final Data

e = Encrypted Data

p = Mathematically verifiable proof

## IX. ADVANTAGES ABOVE EXISTING TECHNOLOGIES

1. Relay groups ensure even if a node breaks down, remaining members of the group can transmit the data.

2. Ensures maximum efficiency, meaning, even if maximum nodes are compromised, still Hydra only requires one honest node in every relay group. Assuming the source/destination are not compromised.

3. If nodes break down in between, entire path will not have to be dissolved and save computational time and reduce latency.

4. Additional mathematical proof ensures the encrypted data is yet not corrupted with re-encryption.

5. Shorter transmitting path as compared to major designs.

6. Employment of the *any-trust* model of operation.

## X. RISKS AND THREATS

1. *Source/Destination Collusion:* Considering the fact that no concept can be 100% flaw-less, Hydra too comes with certain prerequisites, which are that none of the source or destination must already be compromised for Hydra to be successful or they must be colluding with the hacker in any unethical manner. For example, to share encryption keys.

2. *Possible Directory Attacks:* Hydra still utilizes the existence of a maintained directory on a remote server to

store all public keys and addresses of every node active on the network of Hydra. This eventually poses the issue of directory attacks to be executed to compromise the network.

3. *Increased Attacks on Small Networks:* Moreover, if Hydra is run on small networks, then it increases the risk of being compromised because the lesser the number of nodes, the lesser the fingerprints and patterns for communication which can be ultimately figured out by the attackers to ultimately compromise the data and the network, by an extension.

4. *Unnecessary Bandwith:* Not exactly a risk, but more of a disadvantage that Hydra poses is receiving unnecessary useless data at every node and thereby consuming high band-with.

## XI. CASE STUDY

Initially during the first phase of development the idea of elevating the convincing capability of this paper, valid case study evidence was decided to be added for both small scale deployment and large-scale deployment. Venues/environment of the deployment is to be duly kept into consideration while referring to this case study. The entire networksboth in case of small scale and large-scale deployments systems/nodes on the networkwere first analyzed and judged with a pre-installed anonymizing routing mechanism like TOR (The onion router) on the parameters including encryption, corruption of data, data transfer speed and path rebuilding speed.

### *1. Small Scale Deployment – Gamma Testing*

For small scale deployment, a small laboratory of about 10 systems/nodes was finalized and utilized. The network originally having smallnumber of utilized IP addresses was not sub-networked. The original project designed in Python 2.7 consisting of a stand-alone executable script was installed remotely on each system and the data packets were initiated for transfer. For testing purposes, the destination was user inputted. And the data transfer was monitored using network monitoring tools like Wireshark.

The encryption and decryption speeds were monitored and recorded in tabular form. Similarly, various cyber-attacks and sniffing mischiefs were conducted to monitor data corruption possibilities. Finally, denial of service attackwas conducted to remotely shut down nodes by overflowing their input buffers in middle of data transfer in order to monitor path rebuilding speed in both the routing techniques: TOR and Hydra. Finally, the original source code and executable files were released to a group of 50 students to voluntarily use Hydra and again progress of the project was recorded. Finally, the recorded data for both TOR and Hydra was compared and analyzed in order to improve the efficiency of Hydra. During the gamma testing sessions:

a. Path rebuilding speed of Hydra was recorded as incrementally higher than that in TOR in the similar environment with similar conditions.

b. Hydra reported higher chances of node corruption due to limitation of a small network.

c. Hydra's encryption/decryption speed was recorded decrementally lower than TOR due to the presence of an extra mathematical proof along with the original encrypted data. It was, though, found to be not of great demoralizing value to discourage the implementation of Hydra.

### *2. Large Scale Deployment – Beta Testing*

For the purpose of beta testing/large scale deployment, a larger networking laboratory of around 50 systems was utilized. Similarly, as before, original project designed in Python 2.7 consisting of a stand-alone executable script was installed remotely on each system and the data packets were initiated for transfer. For testing purposes, the destination was user inputted. And the data transfer was monitored using network monitoring tools like Wireshark. The encryption and decryption speeds were monitored and recorded in tabular form. Similarly, various cyber-attacks and sniffing mischiefs were conducted to monitor data corruption possibilities and path rebuilding speed in case of a node failure. Further, for beta-testing purpose, the program was publicly released within a group of about 200 students using the same network for the next 5-6 days and constantly the parameters were judged and analyzed for both Hydra and TOR each for 5 days of operation. Complaints and suggestions were noted, considered and executed as deemed fit. During the beta-testing phase:

a. Path rebuilding speed for Hydra was again recorded to be higher by a small difference than TOR.

b. Considering large number of nodes on the network, this time Hydra reported drastically low chances of node corruption/compromise.

c. Hydra's encryption/decryption speeds were found to be vaguely similar to that of TOR when graph lines were plotted.

## XII. CONCLUSION AND FUTURE WORKS

As explained, the Hydra model is a new and considerably better alternative to existing technologies containing major drawbacks. And for future directions, we expect to deploy Hydra over a real network and learn from actual users. We will doubtlessly be able to better evaluate this model after the project is deployed into a much wider network including latency issues, increased performance, compromising-prevention strategies and robustness. Though, gamma testing phase was a little rocky yet successful, we look forward to the next version of Hydra whereinit is planned to employ a

scalable public key distribution by leveraging identity based encryption and eventually multicasting Hydra as an anonymous routing mechanism.

## REFERENCES

[1] Roger Dingledine, Nick Mathewson, and Paul Syverson, "Related Works" in *theirdraft on "Challenges in deploying low-latency anonymity."*

[2] A. Teich, M. S. Frankel, R. Kling, and Y. Lee. Anonymous communication policies for the internet: Results and recommendations of the aaas conference. Information Society, 15(2), 1999.

[3] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In Proceedings of the 13th USENIX Security Symposium, pages 303–320, August 2004. W.-K. Chen, *LinearNetworksandSystems.*Belmont, CA: Wadsworth, 1993, pp. 123–135.

[4] Maheshkumar S.Kamble, Hatkar S.S, " Anonymous Communication in Computer Networks: A Survey, "International Journal of Advanced Research in Computer Science and Software Engineering, , Volume 3, Issue.3, pp.660-664, March 2013

[5] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudo-nyms. Communications of the ACM, 4(2), February 1981.

[6] Roger Dingledine, Nick Mathewson, Paul Syverson, Steven Murdoch, "Attacks and Defense - Passive Attacks, " in their draft "Tor: The Second-Generation Onion Router (2014DRAFTv1)"

[7] Roger Dingledine, Nick Mathewson, Paul Syverson, Steven Murdoch, "Attacks and Defense - Active Attacks, " in their draft "Tor: TheSecond-Generation Onion Router (2014 DRAFTv1)"

[8] Li Zhuang, Ben Y. Zhao, Antony Rowstron in their paper "Cashmere: Resilient Anonymous Routing."

[9] Kun Peng, Juan Manuel, Yvo Desmedt, Ed Dawson in their paper on "Klein Bottle Routing: An alternative to onion routing and mix network."

[10] Kazuya Sakai, Min-Te Sun, Wei-Shinn Ku, Jie Wu in their IEEE paper on "Anonymous Routing to Maximize Delivery Rates in DTNs."

[11] Ariadne in her paper on "A secure On-Demand routing protocol for Ad hoc networks."

[12] S. Le Blond, D. Choffnes, W. Zhou, P. Druschel, H. Ballani, and P. Francis. Towards efficient traffic-analysis resistant anonymity networks. In Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, SIGCOMM '13, pages 303–314, New York, NY, USA, 2013. ACM.

[13] David Easley, Jon Kleinberg in their paper on "Networks, Crowds, and Markets: Reasoning about a Highly Connected World."

[14] Michael J. Freedman, Robert Morris in their paper on "Tarzan: A Peer-to-Peer Anonymizing Network Layer."