International Advance Journal of Engineering Research (IAJER)
Volume 3, Issue 7 (July- 2020), PP 01-10
ISSN: 2360-819X
www.iajer.com

Research Paper **Open Access**

Preserving Security in P2P Networks: A Review

Sidney Burke[1], Ren Zhenya[2]
[1,2]*Department of Computer Engineering, Tallinn University, Tallinn, Estonia*
**Corresponding author: Sidney Burke*

ABSTRACT : Peer-to-Peer networks are powerful distributed applications that use the resources of their own users instead of relying on external servers. These days, privacy-preserving plays an important role for users. As a result, security has become an important subject that should be considered as a priority for these networks. There has been a lot of researches to provide security for P2P users. These methods can be categorized as follows: 1- methods based on mixes, 2- innovative tunneling, 3-broadcast encryption, 4- distributed file systems, 5-Friend-to-Friend, and 6- security for structured P2P networks. The purpose of this paper is to review each category and by giving a practical example for each method, show how they work and what their weaknesses are. Studying these methods can provide a good foundation for available security solutions in P2P networks.

Keywords - P2P, Security, Peer-to-Peer, Computer Networks, Structured, Unstructured.

I. INTRODUCTION

Peer-to-Peer (P2P) is a kind of network that, unlike client-server applications, does not confine itself to a handful of limited servers. These networks, instead of relying on external servers, use resources of their own users. This means that as users get resources from other members, they should also share back with others. Therefore, when new users join the network, the resource supply automatically expands itself, and as they leave, resources also shrink. This means that unlike client-server applications, P2P networks are not fixed with their capacity and adapt themselves to new situations. The scalability of these networks comes from this feature. When new users join or leave (churn) [1], the capacity of the network also becomes bigger or smaller. But compared to client-server applications, the drawback for this method is that the resources are unreliable [2], [3], [4]. The problem comes from resource providers who are regular people that may leave the network at any moment [5], [6].

These networks are mostly used for file-sharing and also provide the technology for many forms of networks [7], [8], [9], [10], [11], [12]. Popular types of P2P applications, such as VOIP [13], [14] and streaming services [14] are attesting for the influence of P2P technology in computer networks. The popularity and the variety of technologies that are based on P2P networks show the utmost importance of the security of users in these networks.

Security is not a problem exclusive to P2P networks. It is also a very important subject in client-server applications, which led to many forms of methodologies. But in P2P networks, we encounter new problems that may never arise in other networking paradigms. For instance, since we do not have a stable infrastructure for our resources, some borrowed methods from client-server applications may cause usability problems for the network. Or connecting thousands of users who do not know each other is a security risk in itself. Considering these problems brings more complications to the security of users for P2P networks [15], [16].

Just like traditional client/server networks, in P2P applications, we have three kinds of anonymity: 1- anonymity of the sender. This can be the person who shares data with others. 2- anonymity of the receiver. The person who wants to download data. 3- The connection between sender and receiver. Protecting connections is required to prevent line spies from reaching to the sender or receiver. Some of the anonymous methods only can secure one of these aspects, but it is clear that the best kind of security is to shield all three aspects against attackers [17].

Currently, methods used for anonymity in P2P networks can be categorized into six different groups: 1- methods based on mixes, 2- innovative tunnelling, 3-broadcast encryption, 4- distributed file systems, 5-Friend-to-Friend, and 6- Security methods for structured networks. The purpose of this paper is to study each of these methods. In this process, we take a practical approach by providing an example for each category.

In Section 2, we talk about methods based on Mixes. Section 3 is about innovative tunnelling. In Section 4, broadcast encryption is presented. Section 5 is for distributed file systems that cover Freenet and

GnuNet. Section 6 is about F2F networks, which is one of the simplest and most efficient methods to provide anonymity. Section 7 is for the security of structured networks. In the end, we conclude this paper in Section 8.

II. METHODS BASED ON MIXES

These networks are based on Mix-Net [18], [19] and use onion routing to provide anonymity. This method is very similar to the original Mix-Net, but the difference here is, we do not have access to external Mix servers. Just like other P2P networks, users are responsible for providing resources and acting as Mixes. The most notable networks in this group are Tarzan [20], MorphMix, and Tap. The three aforementioned networks are very similar in their core operations. They use onion routing as a base, but each takes a different approach to make the tunnels.

The purpose of tunneling in this way is to make anonymous tunnels by making onion routes from a pool filled with thousands of nodes. As a result, messages can be protected with encryption and get transferred anonymously in tunnels. In this method, the sender is the node that wants to make a connection with the receiver by sending a message. To send the message, at first sender has to select a handful of random nodes. And then, a path will be made by using onion routing.

Onion routing uses cryptography to protect data. Each node has an ID number. This ID is usually the IP address of the node. For the sender to make the path, IP addresses of nodes are used as keys for both encryption and decryption. The sender uses these IP addresses and makes a layer of encryption over message for each IP until there are no other intermediate nodes left. As the message travels from sender to receiver, in each step, nodes remove one of the encryption layers and receive the IP address of the next node. This process continues until the receiver gets the message and by decrypting the last layer, obtains the contents of the message. To make this process more clear, we use Figure 1. Sender to send a message for receiver uses nodes A, B, and C as intermediate nodes. In the first layer, the message is encrypted with the IP address of the receiver. For other layers, the IP addresses of C, B, and A are used one after another. For routing, each node uses its IP address as a key to get access to the next IP address until the message reaches the receiver.

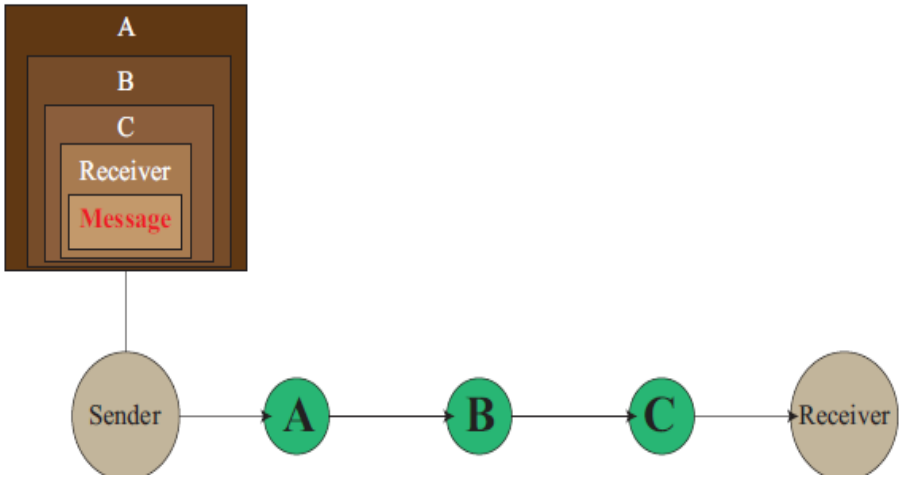

Figure 1: Sender uses onion routing to send messages anonymously for receiver.

Tarzan is the first Mix-Net based P2P network. It is very similar to Tor [21] on how it manages its routing and remains loyal to the original idea of Mix-Net. In this network, the initiator of the message has complete control over choosing intermediate nodes for creating encrypted layers. The problem with Tarzan is that this method lacks scalability and did not adjust itself properly to the P2P environment [20].

To fix these shortcomings, MorphMix takes a different approach. In Morph Mix, tunnels are completed in the middle of the path. Since nodes need to memorize less public keys, it has more harmony with the dynamic nature of P2P networks. Of course, this approach has an obvious obstacle to prevent Sybil attacks. It means that how we can prevent attackers to not introduce themselves as regular nodes. If attackers are successful with Sybil attack, then they can hijack tunnels when building the remaining parts of the tunnel. By doing so, they can peel off all the layers and get access to the message. To prevent this situation, a method is suggested by the creator(s) of Motph Mix, which is called collision detection [22]. But it did not take long to substantiate the inefficiency of this method [23].

The most important feature in TAP, which separates it from the other two networks, is its ability for fault tolerance. A weakness of the design of Tarzan and MorphMix is the lack of stability and longevity of tunnels. The reason which leads to this problem is that nodes can join or leave at any moment, so tunnels can

break at any time. But TAP can provide more stable tunnels than the other two networks. The idea is that Tap does not force messages to route only into one specific path. What TAP proposes is, based on network infrastructure and ID distribution, there should be K alternatives for each step. This means K nodes with the same ID are available in the network. As a result, if one node goes down, there are K-1 nodes to choose from [24]. It should be warned that the security effects of using the same ID addresses for a group of nodes instead of unique IP addresses for each node and its effects on the credibility of onion routing in a distributed environment, has not been addressed with any researches so far.

III. INNOVATIVE TUNNELING

Tunneling includes methods that try to provide anonymity by making a path between the sender and receiver with randomly selected intermediate nodes. Tunneling is a very useful feature that is used in a lot of methods. As we have seen in the previous section, onion routing also uses tunneling to provide anonymity for its users. But there is a dangerous hole in this approach. In these networks, the sender and receiver are trusted entities, and it is other users who attack their transactions. This approach, which is inherited from client-server applications, does not fit into the new world of P2P networks. To clarify this danger, we take a look at a simple attack that can be used easily against users.

To understand how this attack works, we have to first realize how files are found in a distributed network such as Gnutella [25]. To search for a file, a query has to be sent in the network. If it is an unstructured network, the search usually takes place with the flooding algorithm. The simplest form of flooding is that each node asks its neighbors to see if they have the data; the query is looking for. If they don't, the message is forwarded to their neighbors. This process continues until the file is found or messages reach their TTL (time to live) limits. In this way, those nodes which have the file and are willing to share, return their IP addresses to inform that they are ready for upload. After the query has done its research and the original node receives IP addresses, it is time for connecting to those IPs and start to download.

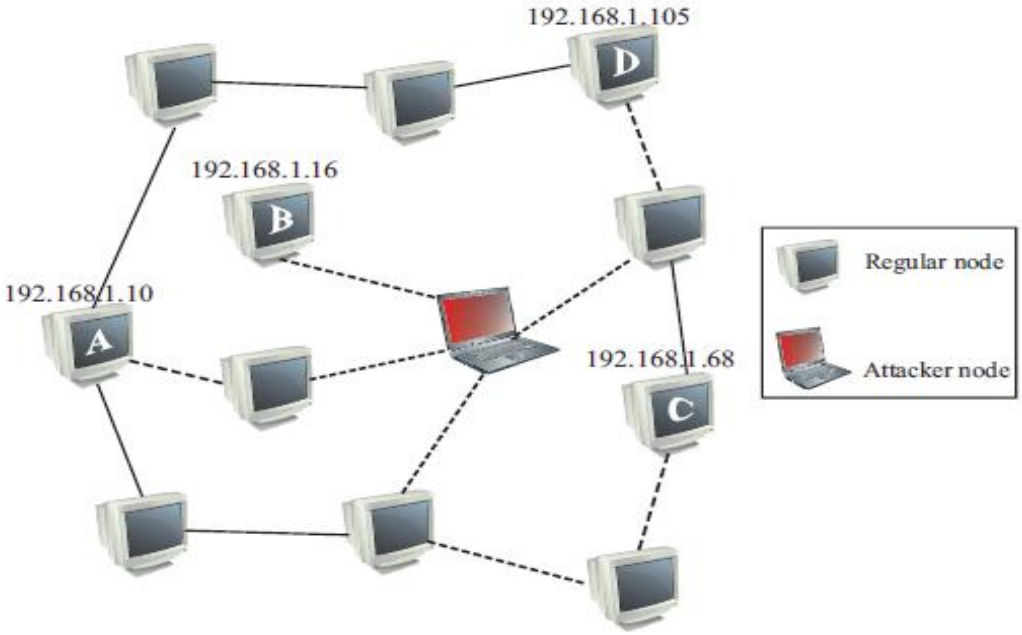

Figure 2: Nodes send their IP addresses which can be used against them.

The problem comes from this fact that the IP addresses of nodes in the network are transparent for all members. Since nodes, to connect with each other, should reveal their IP addresses, attackers by obtaining these IPs, can reveal the identity of users and use it against them. This scheme is shown in Figure 2.

As can be seen, an attack that reveals the identity of users can be made in the simplest form. A security method which neglects to fix this situation and promotes its method as secure can endanger its users to serious damages.

To fix this problem, some innovative methods on tunneling is used to prevent the disclosure of sender and receiver from each other. The most important attempts in this category are probability path, which only provides sender anonymity and Mute network, which can provide mutual anonymity for both the sender and receiver.

IV. PROBABILITY PATH

In this method, for making the path, nodes use a probability algorithm to decide if they want to continue the path or end it by sending the message to the receiver. The most important networks in this group are Shortcut, SSMP, and AP3. Among these three networks, AP3 has the simplest design and is the best example to demonstrate how this method works. Other networks may improve this design by incorporating the probability path with other methods such as onion routing. AP3 uses crowds as its base to provide anonymity in a P2P environment.

Crowds [26] was first introduced to be used as an alternative and a stronger version of proxy servers for client-server applications. In this system, instead of only one proxy, there is a pool of proxy clients. The main idea is that before sending a message to the receiver, we have to get help from other nodes. In this way, the anonymity of the sender is preserved, and it will not be known who sent the original request. This means the attacker can only guess about the identity of the sender but can never be sure about it. An example of this network can be seen in Figure 3. As we can see, all proxies know about the receiver, but the receiver cannot guess who will be the sender. To send a message, the path continues to be made randomly until one node decides to -instead of making another path- contact receiver directly. In this case, D has decided to end the path by sending for the receiver. As a result, the path is made with A, B, C, and D as intermediate nodes.

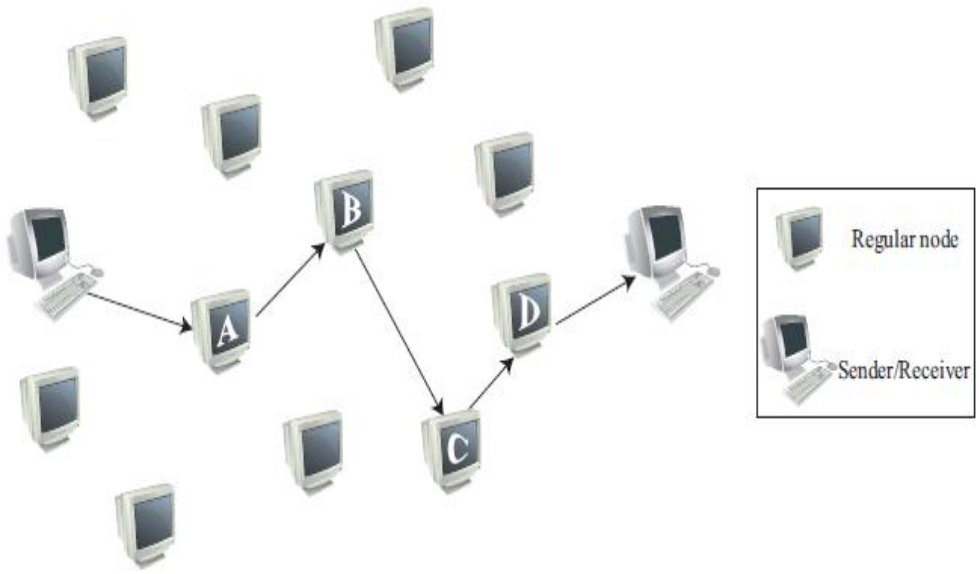

Figure 3: Probability path in Crowds.

Crowds uses a central server (Blender) to manage users of the network. New users, to join the network, have to first connect with Blender. After the first contact, Blender registers a new account for the user. Each user of the network is called a Jundo. Each user, by registering its Jundo in Blender and giving its IP address and port number will be ready to serve its duty as a new proxy for the network.

In AP3 [27], the methodology used in Crowds is combined with Pastry. Just like Crowds, in AP3, we also have to make probability paths. When a node gets a message with a probability value (between 0.5 to 0.9), it decides if it wants to continue the path or send the message to the receiver. If the node decides that path should be continued, then another node will be selected. Otherwise, the message will be transferred directly to the receiver.

By relaying messages between nodes and making a path, the sender remains anonymous from the receiver(and other nodes in the network). But the receiver needs the address of the sender so it can reply to the message. To overcome this problem, just like Crowds, anonymous channels are made to help the receiver. To manage the channel, the sender makes a specific ID for every message it wants to send. Nodes in the path keep this ID so the message can be transferred backward through the path.

As for shortcoming(s) for this method, anonymity is only preserved for senders. Also, since cryptography is not used, it is unable to thwart line spying.

V. MUTE NETWORK

Mute [28] tries to provide mutual anonymity by connecting nodes to each other indirectly. The notion of mute is to use the ant-based algorithm [29] to prevent a direct connection between the sender and receiver. In

this network, all the messages, from query searches to replies for the query, sending the files, etc., goes through the tunnels without any direct connections.

In Mute, each node before connecting to the network receives a unique ID from a hash algorithm like sha1. This ID is used to replace IP addresses. Suppose that we want to find data in the network. Instead of IP addresses, this time, the ID numbers are received. Since there is no way to relate between IP addresses and hash ID numbers, the privacy of users will be preserved. This process is shown in Figure 4.

As can be seen, since there is no direct connection between the sender and receiver, each node can only see its neighbors and can never be sure who has given back its hash ID. In this way, anonymity is preserved not only from intermediate attackers but also protects senders and receivers from each other.

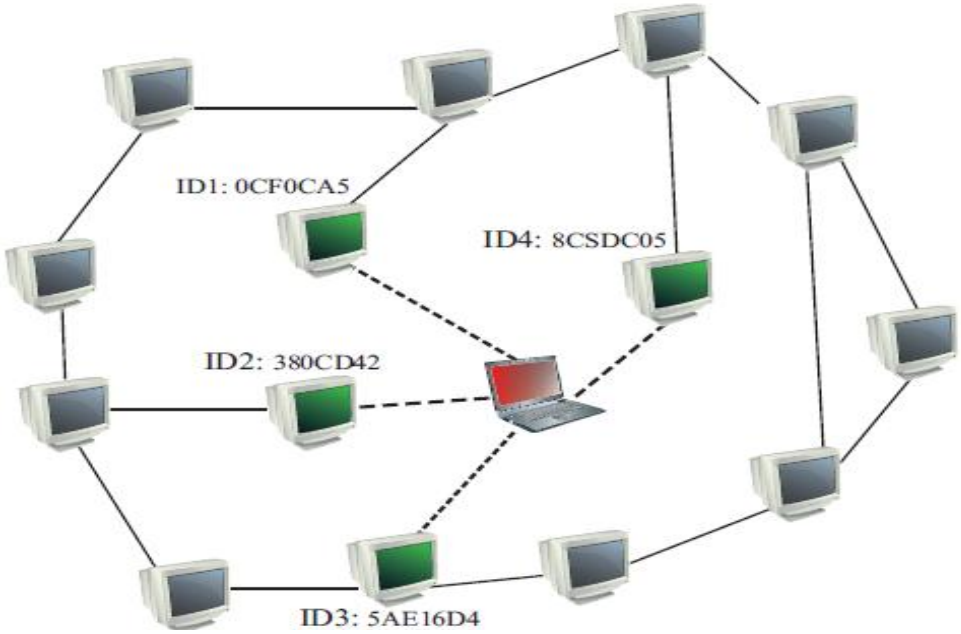

Figure 4: Mute network replaces IP addresses with ID numbers to preserve anonymity.

As we mentioned, in Mute, IP addresses cannot be revealed to other nodes except neighbors. Therefore we need a way to join nodes in the network indirectly. To do this, the ant algorithm can be used for sending data between nodes. The original ant algorithm is replaced with a simpler and more efficient version. This means that, in each query, instead of one sign (pheromone), two signs are used.

The first sign is required so messages can go back to the sender of the query, and the second sign is to help wandering queries. In this algorithm, queries roam in the network sporadically to find nodes with desired data. Every time a query visits a node, the first sign will be registered. In this way, after data is found, messages can transfer back to the query's initiator. In their way back to the initiator, the second sign is used for each node in the path. This helps to inform the wandering queries, that data is found so they can use the right path.

VI. BROADCAST ENCRYPTION

Broadcast encryption is one of the most powerful methods to preserve anonymity. But its biggest hurdle is the lack of efficiency and scalability. This method first appears in DC-Net and then came to P2P networks with Herbivore and P^5. Between these two networks, only Herbivore has a real-world application, and P^5 is only implemented as a simulation environment. The notion of this method is to send encrypted messages for all nodes in the network. But since only the receiver has the key to decrypt messages, the anonymity of the receiver and their connection are preserved. For the sender to also preserve its anonymity, all nodes constantly should send dump messages, so it is not clear when a real message is going to be sent.

In P2P, just like other networks of this family, channel(s) are used to send data. All the messages have the same lengths and include real and dump messages. These messages are encrypted with the public key of the receiver. As a result, only the receiver with its private key is able to decrypt them. In current broadcast solutions for P2P networks, nodes connect with each other in a simple ring. In the next step, these rings connect to each other in a more sophisticated infrastructure to help the scalability of the network. For this purpose, P^5 separates these channels and then uses a hierarchical tree structure to arrange these rings [30]. Herbivore, instead, divides the network into smaller cliques. These cliques connect with each other in a global ring. This is

done by getting help from the design of Pastry [31].

VII. USING DISTRIBUTED FILE SYSTEMS TO ACHIEVE PRIVACY

The purpose of the networks based on this method is to make a big distributed file system and spread data in this big hard drive. As it is shown in Figure 5, File1 is divided into three parts and distributed in the network. If node A wants to retrieve File1, it has to search in the network and get packets one by one. By getting help from this infrastructure, the security of data, anonymity of the sender, and anonymity of the receiver can be preserved.

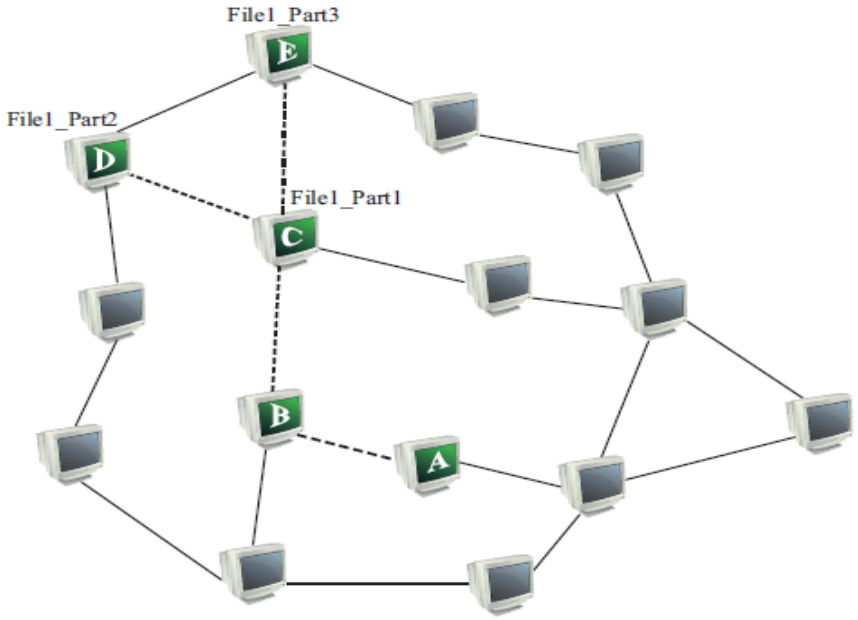

Figure 5: Distributed file systems can be used as a base infrastructure to provide anonymity.

This method is very powerful, but there is a weakness in its design. A distributed file system is more suitable for stable servers. This is in complete contrast with P2P environments in which nodes do not guarantee to remain in the network. This contradiction usually results in an unsatisfying experience for users and tends to slow down the network. Three main networks that use this method are GnuNet, Freenet, and Entropy. Among these three, Entropy is discontinued since 2004, but the other two are still under active development.

VIII. FREENET

Ian Clarc has first introduced Freenet in 1999 as an anonymous P2P network [32]. In Freenet, the whole network is treated as a cohesive hard disk. This hard disk is made by chaining together the resources that users donate to the network. It means that each user to help the network contributes a part of its own space. For data to be shared, at first, it has to be divided into smaller pieces, and then those pieces spread in that distributed file system. Since all the data are encrypted, no one is aware of the data he is sharing with others. In this way, the security of the person who is sharing with others is preserved. On the other hand, for downloading data, Freenet gets help from tunneling techniques, so the anonymity of the receiver is also preserved. For the person who shares the data, since after doing it, he can leave the network, it never is revealed who is the first uploader. So his anonymity is also preserved.

Every data in the network has a specific key, and each key has a specific place. For routing, instead of flooding, a simple greedy algorithm is used. In this way, if a node does not have the key and has to choose between its neighbors, a neighbor with the key, most similar to the searching key is selected.

To retrieve packets, just like Mute, there are no direct connections, and intermediate nodes are used to make tunnels. Therefore, after a search query reaches a node, it marks that query in its database. To retrieve the data, these marks are used to send packets back to the node which sent the original query.

IX. GNUNET

GnuNet is a distribute P2P file-sharing network. It is specifically designed to provide anonymity for users. This network uses cryptography to shield data and GAP [33] for anonymous routing. Just like Freenet, we can consider it as a big distributed file system.

In GnuNet, there are two kinds of messages, queries, and replies. The query messages are used to find data, and reply messages are used to transfer them back. Unlike the tunneling method used in Freenet, tunnels do not get specific paths in GnuNet. Each node decides if it wants to be included in the path or not. This process takes place in an intelligent way. Nodes can recognize the workloads on the network. If traffic is too much, a node may not mark itself as a path. But if traffic is light, it marks itself as part of the tunnel. Also, nodes are able to emulate the functionality of Mix-Net by grouping different messages together and using delays. In this way, observers have more difficulties in analyzing the network. To protect data from identifying and make spying a more difficult job, GnuNet prevents a specific place for packets. This means that packets have the ability to move from one node to the other.

X. FRIEND-TO-FRIEND

This is a very simple and yet one of the most effective methods used for anonymity in P2P networks. The main idea is to only let trusted people engage in transactions, and unauthorized people are prevented from entering the network. This method is usually combined with other P2P networks. For instance, Darknet in Freenet or OneSwarm [34] for BitTorrent protocol [35] are all examples of this method. Automating the analysis of user behavior and blocking malicious users also belongs to this category. For this purpose trust management systems [36], [37], [38], [39] and machine learning tools [40], [41] which use unsupervised learning [42] for analyzing user behavior is recommended.

XI. STRUCTURED OVERLAYS

The methods described were exclusive to unstructured overlays that use flooding algorithms in their infrastructures. The flooding algorithms cannot guarantee to find rare objects. The reason is that these methods need time-to-live counters, which may go to zero before finding the object. To solve this problem, structure P2P systems such as Chord [43], CAN [44], Pastry [45], and Tapestry [46] use distributed hash tables (DHT) to provide routing mechanisms. This routing mechanism replaces the flooding algorithms and does not have any problem to find objects. The routing in these systems is based on the contents of the networks and not the nodes. Therefore, users connect with each other in relation to their contents.

Providing security for Structured overlays is especially difficult because in each step DHT table requires the identity of the destination. If for security purposes, the destination is hidden, it blocks the routing mechanism [47]. Because of the complete change of the routing mechanisms, most of the methods in unstructured networks cannot be implemented for these networks. To mitigate this problem, the Agyaat [47], proposes a new infrastructure for mutual anonymity for structured P2P networks. To overcome the aforementioned problems with DHT tables, they combined structured and unstructured environments to create a semi-structured overlay. The other methods, such as [48], [49] try to make identification harder for the middle nodes that provide security. The papers [50], [51] provide anonymity without relying on unstructured overlays by using the intrinsic features (geometrical shapes) of the structures overlays. These methods can be applied to the networks that have a Circular topology, which is first introduced in Chord [43]. There are also several attempts that improve the circular design such as SelfChord [52], F-Chord [53], BSRE [54], and Hybrid-Chord [55]. In all of these designs, their circular shapes can be used to provide anonymity for the users. The provided anonymity in most cases can get further strengthened with intelligent systems [56] and machine learning methods [57].

XII. CONCLUSION

Preserving the privacy of users and keeping their anonymity is a very important issue in P2P networks. Neglecting to take action in this matter may lead users to serious problems. In this paper, methods that provide security for P2P networks are categorized, and at least one network in each group has been reviewed. Each method has its pros and cons, but there are some common problems that can be seen in a lot of solutions. Researchers should recognize these problems and try to prevent them in their prospective designs. These issues are as follows: 1- we have to understand that there are fundamental differences between P2P networks and client-server applications. As a result, each architecture may have its own specific attacks that cannot be applied to the other architecture. Some researchers tried to utilize methods used in client-server applications without optimizing them for P2P networks. 2- Performance is a very important factor that should be considered as a priority in design decisions. In P2P networks, users share their own resources with others, so they experience an overload that users of other networks may not experience. Researchers should always consider this fact and not worsen the situation by bringing extra loads for their methods. 3- It should always be considered that P2P has a dynamic environment. In these networks, nodes may leave and join at any moment. If a network with strong anonymity asks its users to stay in the network for days so it can stabilize their environment, it may undermine its strong security and alienate users from its secure environment.

REFERENCES

[1]. Yang, D., Zhang, Y.X., Zhang, H.K., Wu, T.Y. and Chao, H.C., 2009. Multi-factors oriented study of P2P Churn. International Journal of Communication Systems, 22(9), pp.1089-1103.

[2]. Berenjian, S., Hajizadeh, S., Hatamian, M. and Atani, R.E., 2019. An Incentive Security Model to Provide Fairness for Peer-to-Peer Networks. arXiv preprint arXiv:1906.09355.

[3]. Naghizadeh, A., Razeghi, B., Radmanesh, I., Hatamian, M., Atani, R.E. and Norudi, Z.N., 2015, April. Counter attack to free-riders: Filling a security hole in BitTorrent protocol. In 2015 IEEE 12th International Conference on Networking, Sensing and Control (pp. 128-133). IEEE.

[4]. Naghizadeh, A., 2016. Improving fairness in peer-to-peer networks by separating the role of seeders in network infrastructures. Turkish Journal of Electrical Engineering & Computer Sciences, 24(4), pp.2255-2266.

[5]. Buford, J., Yu, H. and Lua, E.K., 2009. P2P networking and applications. Morgan Kaufmann.

[6]. Qureshi, A., Megías, D. and Rifà-Pous, H., 2015. Framework for preserving security and privacy in peer-to-peer content distribution systems. Expert Systems with Applications, 42(3), pp.1391-1408.

[7]. Wang, Q., Wang, J., Yu, J., Yu, M. and Zhang, Y., 2012. Trust-aware query routing in P2P social networks. International Journal of Communication Systems, 25(10), pp.1260-1280.

[8]. Mavromoustakis, C.X., 2013. Mitigating file-sharing misbehavior with movement synchronization to increase end-to-end availability for delay sensitive streams in vehicular P2P devices. International Journal of Communication Systems, 26(12), pp.1599-1616.

[9]. Amad, M., Meddahi, A., Aïssani, D. and Vanwormhoudt, G., 2012. GPM: A generic and scalable P2P model that optimizes tree depth for multicast communications. International Journal of Communication Systems, 25(4), pp.491-514.

[10]. Kryftis, Y., Mavromoustakis, C.X., Mastorakis, G., Pallis, E., Batalla, J.M. and Skourletopoulos, G., 2014, December. Resource usage prediction for optimal and balanced provision of multimedia services. In 2014 IEEE 19th international workshop on computer aided modeling and design of communication links and networks (CAMAD) (pp. 255-259). IEEE.

[11]. Mavromoustakis, C.X. and Karatza, H.D., 2011. Embedded socio-oriented model for end-to-end reliable stream schedules by using collaborative outsourcing in MP2P systems. The Computer Journal, 54(8), pp.1235-1247.

[12]. Palazzi, C.E. and Bujari, A., 2012. Social-aware delay tolerant networking for mobile-to-mobile file sharing. International Journal of Communication Systems, 25(10), pp.1281-1299.

[13]. Zhang, G. and Fischer-Hübner, S., 2019. A survey on anonymous voice over IP communication: attacks and defenses. Electronic Commerce Research, 19(3), pp.655-687.

[14]. Ijaz, H., Welzl, M. and Jamil, B., 2019. A survey and comparison on overlay-underlay mapping techniques in peer-to-peer overlay networks. International Journal of Communication Systems, 32(3), p.e3872.

[15]. Li, B., Erdin, E., Gunes, M.H., Bebis, G. and Shipley, T., 2013. An overview of anonymity technology usage. Computer Communications, 36(12), pp.1269-1283.

[16]. Romero-Tris, C., Castellà, D., Viejo, A., Castellà-Roca, J., Solsona, F. and Mateo-Sanz, J.M., 2015. Design of a P2P network that protects users' privacy in front of Web Search Engines. Computer Communications, 57, pp.37-49.

[17]. Pfitzmann, A. and Waidner, M., 1987. Networks without user observability. Computers & Security, 6(2), pp.158-166.

[18]. Chaum, D.L., 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 24(2), pp.84-90.

[19]. Hatamian, M., Barati, H., Movaghar, A. and Naghizadeh, A., 2016. CGC: centralized genetic-based clustering protocol for wireless sensor networks using onion approach. Telecommunication systems, 62(4), pp.657-674.

[20]. Freedman, M.J. and Morris, R., 2002, November. Tarzan: A peer-to-peer anonymizing network layer. In Proceedings of the 9th ACM conference on Computer and communications security (pp. 193-206).

[21]. Dingledine, R., Mathewson, N. and Syverson, P., 2004. Tor: The second-generation onion router. Naval Research Lab Washington DC.

[22]. Rennhard, M. and Plattner, B., 2002, November. Introducing MorphMix: peer-to-peer based anonymous Internet usage with collusion detection. In Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society (pp. 91-102).

[23]. Tabriz, P. and Borisov, N., 2006, June. Breaking the collusion detection mechanism of MorphMix. In International Workshop on Privacy Enhancing Technologies (pp. 368-383). Springer, Berlin, Heidelberg.

[24]. Zhu, Y. and Hu, Y., 2004, August. Tap: A novel tunneling approach for anonymity in structured p2p systems. In International Conference on Parallel Processing, 2004. ICPP 2004. (pp. 21-28). IEEE.
[25]. The gnutella protocol specification v0.4. URL: http://stanford.edu/class/cs244b/gnutella_protocol_0.4.pdf
[26]. Reiter, M.K. and Rubin, A.D., 1998. Crowds: Anonymity for web transactions. ACM transactions on information and system security (TISSEC), 1(1), pp.66-92.
[27]. Mislove, A., Oberoi, G., Post, A., Reis, C., Druschel, P. and Wallach, D.S., 2004, September. AP3: Cooperative, decentralized anonymous communication. In Proceedings of the 11th workshop on ACM SIGOPS European workshop (pp. 30-es).
[28]. MUTE: Simple, Anonymous File Sharing. URL: http://mute-net.sourceforge.net/
[29]. Naghizadeh, A., Berenjian, S., Margolis, D.J. and Metaxas, D.N., 2020. GNM: GridCell navigational model. Expert Systems with Applications, 148, p.113217.
[30]. Sherwood, R., Bhattacharjee, B. and Srinivasan, A., 2005. P5: A protocol for scalable anonymous communication. Journal of Computer Security, 13(6), pp.839-876.
[31]. Goel, S., Robson, M., Polte, M. and Sirer, E., 2003. Herbivore: A scalable and efficient protocol for anonymous communication. Cornell University.
[32]. Clarke, I., Sandberg, O., Wiley, B. and Hong, T.W., 2001. Freenet: A distributed anonymous information storage and retrieval system. In Designing privacy enhancing technologies (pp. 46-66). Springer, Berlin, Heidelberg.
[33]. Bennett, K. and Grothoff, C., 2003, March. GAP–practical anonymous networking. In International Workshop on Privacy Enhancing Technologies (pp. 141-160). Springer, Berlin, Heidelberg.
[34]. Isdal, T., Piatek, M., Krishnamurthy, A. and Anderson, T., 2010. Privacy-preserving p2p data sharing with oneswarm. ACM SIGCOMM Computer Communication Review, 40(4), pp.111-122.
[35]. Cohen, B., 2003, June. Incentives build robustness in BitTorrent. In Workshop on Economics of Peer-to-Peer systems (Vol. 6, pp. 68-72).
[36]. Dellarocas, C., 2003. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. Management science, 49(10), pp.1407-1424.
[37]. Naghizadeh, A., Razeghi, B., Meamari, E., Hatamian, M. and Atani, R.E., 2016. C-trust: A trust management system to improve fairness on circular P2P networks. Peer-to-Peer Networking and Applications, 9(6), pp.1128-1144.
[38]. Selcuk, A.A., Uzun, E. and Pariente, M.R., 2004, April. A reputation-based trust management system for P2P networks. In IEEE International Symposium on Cluster Computing and the Grid, 2004. CCGrid 2004. (pp. 251-258). IEEE.
[39]. Alkharji, S., Kurdi, H., Altamimi, R. and Aloboud, E., 2017, November. AuthenticPeer++: a trust management system for P2P networks. In 2017 European Modelling Symposium (EMS) (pp. 191-196). IEEE.
[40]. Chang, S. and Daniels, T.E., 2009, November. P2P botnet detection using behavior clustering & statistical tests. In Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence (pp. 23-30).
[41]. Junior, G.P.S., Maia, J.E.B., Holanda, R. and de Sousa, J.N., 2007, July. P2p traffic identification using cluster analysis. In 2007 First international global information infrastructure symposium (pp. 128-133). IEEE.
[42]. Naghizadeh, A. and Metaxas, D.N., 2018, December. Meaningful Distance for Multivariate Clustering. In 2018 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1149-1154). IEEE.
[43]. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F. and Balakrishnan, H., 2001. Chord: A scalable peer-to-peer lookup service for internet applications. ACM SIGCOMM Computer Communication Review, 31(4), pp.149-160.
[44]. Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S., 2001, August. A scalable content-addressable network. In Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (pp. 161-172).
[45]. Rowstron, A. and Druschel, P., 2001, November. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing (pp. 329-350). Springer, Berlin, Heidelberg.
[46]. Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D. and Kubiatowicz, J.D., 2004. Tapestry: A resilient global-scale overlay for service deployment. IEEE Journal on selected areas in communications, 22(1), pp.41-53.
[47]. Singh, A., Gedik, B. and Liu, L., 2006. Agyaat: Mutual anonymity over structured p2p networks. Internet Research, 16(2), pp.189-212.

[48]. Borisov, N. and Waddle, J., 2005. Anonymity in structured peer-to-peer networks. Computer Science Division, University of California.

[49]. Nambiar, A. and Wright, M., 2006, October. Salsa: a structured approach to large-scale anonymity. In Proceedings of the 13th ACM conference on Computer and communications security (pp. 17-26).

[50]. Naghizadeh, A., Berenjian, S., Meamari, E. and Atani, R.E., 2016. Structural-based tunneling: preserving mutual anonymity for circular P2P networks. International Journal of Communication Systems, 29(3), pp.602-619.

[51]. Naghizadeh, A., Berenjian, S., Razeghi, B., Shahanggar, S. and Pour, N.R., 2015, January. Preserving receiver's anonymity for circular structured P2P networks. In 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC) (pp. 71-76). IEEE.

[52]. Forestiero, A., Mastroianni, C. and Meo, M., 2009, May. Self-chord: A bio-inspired algorithm for structured P2P systems. In 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (pp. 44-51). IEEE.

[53]. Cordasco, G., Gargano, L., Negro, A., Scarano, V. and Hammar, M., 2008. F-Chord: Improved uniform routing on Chord. Networks: An International Journal, 52(4), pp.325-332.

[54]. Naghizadeh, A. and Shahbahrami, A., 2017. Binary search routing equivalent (BSRE): a circular design for structured P2P networks. Transactions on Emerging Telecommunications Technologies, 28(4), p.e3012.

[55]. Flocchini, P., Nayak, A. and Xie, M., 2004, December. Hybrid-chord: A peer-to-peer system based on chord. In International Conference on Distributed Computing and Internet Technology (pp. 194-203). Springer, Berlin, Heidelberg.

[56]. Berenjian, S., Shajari, M., Farshid, N. and Hatamian, M., 2016, September. Intelligent automated intrusion response system based on fuzzy decision making and risk assessment. In 2016 IEEE 8th International Conference on Intelligent Systems (IS) (pp. 709-714). IEEE.

[57]. Naghizadeh, A. and Metaxas, D.N., 2018, December. Meaningful Distance for Multivariate Clustering. In 2018 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1149-1154). IEEE.