Peer-to-Peer Network Security Issues and Analysis: Review

Adel R. Alharbi and Amer Aljaedi

College of Computing & Information Technology, University of Tabuk, 71491, Saudi Arabia

Summary
Peer-to-Peer technologies offer promising and potential features for scalable communication networks. The peer-to-peers decentralized nature along with its extreme flexibility make them a robust solution to an expected network failure since it can avoid the centralized choke points and collectively distribute the network workload along with it vast routing capabilities. Unfortunately, many of the peer-to-peer networks are still immature and suffer from security weaknesses and threats. Among these are: no standardized trust mechanisms, no standard P2P-based API for security operations, including encryption, authentication, and even logging. This paper seeks to present and explore the security weaknesses and threats in the peer-to-peer networks with surveying solutions that are being considered in the real life peer-to-peer network designs and the in networking literature. Finally, we highlight what we identify as a fundamental peer-to-peer network problem: trust of peers and secure traffic routing.

Key words:
Peer-to-Peer Technologies, Network Security, Worms and Virus, Weaknesses, Threats, Attacks, Countermeasures, Detection Systems

1. Introduction

Nowadays, Peer-to-Peer (P2P) technologies offer many advances for computing environment. P2P are often out-lined as sharing of computing resources and services among participants by direct exchange. P2P clients collectively participate in direct sharing of data and processing time. P2P network participant acts as client and server at the same time. Figure 1 shows a simple topology of P2P network.

Such network technologies offer a myriad of solutions to new and traditional network problems. P2Ps decentralized nature, along with its extreme flexibility make them a robust solution to a possible network failure. Unfortunately, many of the P2P networks are still immature and suffer from many security weaknesses and threats. Although there are many solutions have been proposed to the P2P problems but very of them are standardized. One of the problems that we have encountered while surveying the P2P networking issues was the sheer number of different competing P2P technologies available. P2P networks have been utilized

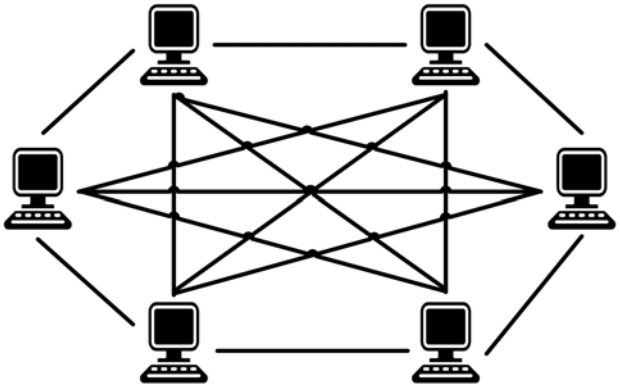

Fig 1. Peer-to-Peer Network

for: IP telephony, file sharing, network file storage, network intrusion detection, instant messaging and many others. Intertwined with these are also a number of security issues that are unique to each solution [1].

With such a broad field of different approaches towards security in different P2P technologies, we have decided to take threefold approach to better serve our survey targeting security issues within P2P networks. First, we introduce the concepts and standards of security in P2P, where we focus on, and we also present solutions to different types of attacks directed against those solutions provided by researchers. Second, we look at how various P2P protocols have been implemented to provide protection. Third, we examine what we have identified as a fundamental problem with today's P2P networks: trust identity management and secure routing. Trust and secure routing become important factors in a P2P network since there is not established hierarchy of nodes. All nodes are by definition peers. Any node could potentially be a malicious node or seek to impact the network in some ways, which raises many security concerns in P2P.

The rest of the paper is organized as follows: Section 2 discusses main security issues in P2P technologies. Section 3 describes major security vulnerabilities associated with P2P networks with the proposed solutions in the literature. Section 4 elaborate on popular P2P implementations. Section 5 defines reputation, trust, and secure routing architecture for P2P networks. Section 6

Manuscript received November 5, 2020.
Manuscript revised November 20, 2020.
https://doi.org/10.22937/IJCSNS.2020.20.11.10

provides the concluding remarks and discusses the prospects for future work.

2. Security Issues In P2P Technologies

P2P applications and services such as KaZaA, Napster or Gnutella become extremely popular over past years because they virtually offered freely resource sharing services among its peers. These P2P technologies represent a paradigm shift from the usual server-client model. In pure P2P networks, there is no server-client architecture. Every note in the network acts as a peer, where there is no hierarchy of systems. Among the vast number of peers, objects (such as documents, songs, movies, voice, chat, etc.) can be widely and freely replicated and distributed, hereby providing resource sharing scalability despite the lack of centralized infrastructure, and the opportunity for high availability [2].

Unlike the earlier systems, new P2P overlays have been designed with more intensive analysis and carefully design to ensure network efficiency and scalability such as: Chord, CAN, Tapestry, and Pastry. Those overlays will offer a self- organizing substrate for large-scale P2P applications. They also provide a strong platform for the construction of decentralized services such as content distribution, network storage, web caching, searching and indexing, and application level multicast. Structured overlays enable applications to find any object in a very probabilistically delimited, small range of network hops, whereas requiring per-node routing tables with exclusive a slight range of entries. Furthermore, these P2P systems are fault-tolerant, scalable, and supply effective load balancing [2].

Deploying measures in such P2P networks can be challenging, as they must be strong against conspiracy of nodes to attack the remained of nodes, including a false and malicious response by a malevolent node, which could either return false routes or false data to a query. Attackers may need variety of alternative goals as well as censorship against systems that attempt to offer high availability and traffic analysis against systems that attempt to offer anonymous communication [2].

The Open Systems Interconnection (OSI) is coupled with security architecture, which it reference model serves as a useful framework to overcome security problems not only in P2P networks. The set of imminent security services are divided into the following classes:

- Confidentiality
- Integrity
- Access Control
- Authentication: containing entity authentication and data origin authentication
- Non-repudiation

In addition, services not explicitly included in the above list could be added. In P2P networks, anonymity and accountability represent some of these services that inherited from the P2P network structure. Depending on the type of scenario or the network, a successful implementation of secure system could require all or some of those varied security services.

For traditional client-server systems, users are usually identified with a user account and system-specific controls are mounted on these accounts to enforce security mechanisms. Security services like accountability and access control is enforced during this manner, with the accounts that represent a user stable identity. Alternative services like non-repudiation and authentication clearly depend upon the institution and preservation of reliable identity management and access control [3]. This concludes that almost all of the generic security services are dependent on the supply of stable identities. Thus, despite the dearth of centralized infrastructure being P2Ps virtue, it also brings additional security challenges for P2P networks.

3. Security Vulnerabilities, Attack Vectors, and Countermeasures

This section introduces some specific security vulnerabilities and attack vectors alongside with proposed solutions. The choice of the specific attacks presented was based on the fact that they are common and well known to P2P systems.

3.1 Distributed Denial of Service Attack (DDoS)

The distributed nature of P2P networks and applications is certainly prone to a flooding attacks such as the Distributed Denial of Service (DDoS). In the DDoS, the attacker exploits an oversized range of hosts, typically referred to as zombies to simultaneously send overwhelmed packets to the target, victim host as it is shown in figure 2. The consequences of such an attack for a victim are severe. DDoS overwhelms the target system with bogus requests and diminish the target capability, which consequently impacts its services. In fact, from an attackers perspective, an effective attack will not only debilitating the target host' resources such as: upstream bandwidth, downstream bandwidth, networking stack, memory, and CPU, it will also involve a large number of zombies from totally different Internet Service Providers

(ISP), which could defeat network-based security controls [4].

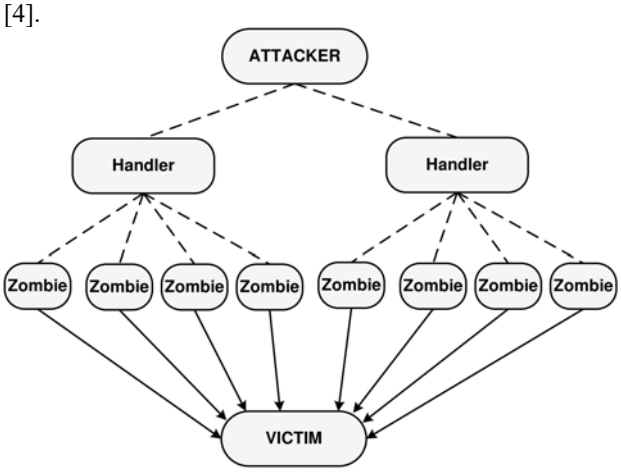

Fig 2. DDoS Attack

Any P2P system that has innumerable of active peers can be used and could serve as a flooding engines for attacks against targeted hosts. Two approaches can be used to create such a DDoS engine. The two categories of flooding DDoS attack, namely: TCP connection attack, which overwhelms victims connection resources with fully-open TCP connections. Therefore, depriving legitimate users from creating connections to victim host. The second category is a bandwidth DDoS attack, which generates volumatic traffic to congest the victims network links. Moreover, there are two types of P2P network poisoning. The first one is poisoning the distributed index within the peers. Second one is poisoning the routing tables within the peers. For each poisoning attack, the targeted host does not need to be a participant of P2P system, it might be a mail server, web server, or maybe a user desktop. A study by Naoumov et al. [5] shows both of these attacks within the Overnet, a popular DHT-based P2P file-sharing network. They argued that using a limited poisoning attack within a short period can exploit Overnets routing tables and indexing, and it would lead to a DDoS attack against targeted host. The results concluded that with modest efforts, each DDoS attacks could direct a big traffic from numerous peers towards chosen target. P2P substrates and index will doubtless be exploited for DDoS attack. The countermeasures to the DDoS attacks are:

- In routing poisoning: A peer *K* receives a message that indicates the existence of a node *G*, whereas *G* is the victim and not a participant within the P2P network. A countermeasure for this problem is that the peer *K* should check if *G* could be a peer within the P2P network.
- In the index attack: An indexing peer *K* receives a broadcast message that advertises the existence of a file at a location *G*. A way to a counter such an attack on a non-participating host is to make peer *K* verify that *G* could be a participant within the P2P network. This will once more be done by having *K* handshake with *G*. However, acknowledgement with every peer *K* that advertises content to *G* might incur considerable overhead of additional network traffic between each pair of *K* and *G*.

3.2 Denial of Service Attack (DoS)

P2P systems are susceptible to denial-of-service attack, which ranges from low-level packet flooding to high level abuse of the communication protocols. The DoS like SYN-flooding that quickly degrade the utility of the victim system [6]. An equal concern should be brought to another dangerous DoS type of attack on P2P services, also named service attrition. The term attrition is used to indicate that the attack slowly waste the victim resources over an extended amount of time, impairing its perform. Maniatis et al. [6] proposed a broad variety of techniques that support resisting attrition attacks on P2P networks with establishing synergism among them. Adversaries can be classified based on the intent and characteristics of their attacks on the victim system:

- Stealth: adversaries try to manipulate, subvert or otherwise compromise the integrity of the content or service functionality of the system in undetected manner. In a file system, this opponent could obtain modification access unobtrusively while not authorization.
- Theft: adversaries try to access restricted zone if the system services to unauthorized copy data.
- Nuisance: adversaries try and cause several apparently false alarms to discredit intrusion detection and observance systems.
- Free-loader: adversaries try to have the benefit of the system's services whereas conducive nothing reciprocally.
- Attrition: adversaries try to stop clients of the system from getting timely service.

To concentrate on the attrition adversary, we define three potential modes that represent the sophistication of attrition attack in exploiting victim's system [6]:

- Pipe stoppage: Through generating huge bogus network traffic, the adversary saturates the victim peer's network links to prevent them from receiving or sending valid messages.
- Anomalously high rates of requests: with sufficient knowledge of the target protocols' operation, an attrition attack could exploit the protocol's requests to send crafted requests to victim at a rate that saturate network connections towards the victim and deplete its resources. Such

clearly abnormal traffic rates could lead to the identification of the attacking agents, and eventually will be stemmed the tide over days interval (e.g., with subsequent filtering and packet marking).

- Seemingly innocuous rates of requests: the adversary transmits requests at lower rates (not greater than the expected rate from the related loyal peers). These requests are crafted and looks regular, but can exhaust the victims' resources or at least disturb the services.

The countermeasures to DoS/Attrition attacks are:

- Workload Balancing: If the effort required by a requester to request from a provider is a smaller than the effort required by the supplier to fulfil the requested services, then the system are often susceptible to attrition attack that consists of large numbers of bogus service requests. An attempt to balancing the workload on the consumed computing resources can help in this situation. On the other hand, if a peer adversary issues an inexpensive request for service, then he might cause the provider to allocate resources that are actually not used and are only free when their timeout run.
- Rate Limitation: Peers ought to satisfy requests "no quicker then necessary" rather than "as quick as possible".
- Admission control: Dropping or rejecting unacknowledged incoming requests.
- Redundancy: Recognize flooding and repetitive requests.
- Compliance enforcement: Proving that the requested service from the supplier has been actually performed after the verification and evaluation of the request.
- De-synchronization: P2P system designer ought to solely choose synchronization if it's necessary; accidental synchronization ought to be prevented by turn-taking, back-off, randomization, and etc.

3.3 Intrusion Attack - Worms and Virus

In the computer networks world worms and viruses are just a simple fact of life. The spreading of worms and viruses is usually involved with intrusion attacks. In a P2P network, the major factor for spreading worms and viruses is increased popularity of freely P2P file-sharing networks. The possibilities of a worm having large scale impact in a P2P network is becoming a reality. With large sets of machines on the network often running homogeneous clients, a worm exploiting vulnerability could in fact compromise large numbers of hosts. These hosts could then be used to capture end user's sensitive information or be used for further attacks, such as Distributed Denial of Service (DDoS) [7]. The need to identify and detect malicious nodes in a network domain becomes exceedingly important.

Concise policy should be used to prevent illegal distribution and downloading of pirated materials , which participate in the viruses spreading. Enforcement of such a policy should be in place although the actual enforcement can prove to be a complex and difficult task, especially in P2P networks. The basic policy to prevent the download and distribution of copyrighted materials is to block the well-known ports that are utilized by P2P software, although the overwhelming majority of P2P software packages (e.g., Bit Torrent) upload and download items of files from different sources on different ports. In addition, there can be considerable number of false positives IDS alerts generated when the peers utilize well-known ports for the P2P sharing.

There are proposed solutions in the computer networks literature to this problem to detect P2P network-based activities and the associated traffic. However, such approach poses a large overhead and presents a delay, if every network packets needs to be deeply examined. Also, there are commercial products available that address the above issues, but those might be expensive (i.e., annual license for updates) and inflexible for a campus community, where various research experiment are conducted. Therefore, D. Ennis et al. [8] proposed a model that combines a several open-source solutions based on policy enforcement framework. Such a customizable model is flexible and can suite different environments.

3.4 P2P Policy Infringement within LAN

An interesting discussion evolves around establishing a security policy for P2P systems in small- or medium-scale environments such as campus LAN. Interestingly enough, it is often in such P2P networks malicious activates take place. Thus, the use of P2P software is one among the current topics under discussion in educational environment these days as the P2P networks can pose a problem for campus communities. Aside from being bandwidth-eaters, P2P software is sometimes used to download and distribute copyrighted materials that could have important legal and monetary implications on campus community.

Clearly and concisely policy should be used to prevent illegal distribution and download of copyrighted materials; enforcement of such a policy should be in place as well. Although the actual enforcement measure can prove to be a complex and difficult task. The elementary policy to prevent the download and distribution of copyrighted materials is to block the well-known ports that are employed by P2P software although the vast majority of P2P software (e.g., Bit Torrent) download and upload parts of files from completely different sources on different

ports. In addition, there can be false positives alerts as result of utilizing well-known ports by the P2P file sharing services [9].

A proposed solution to the above problem is to discover P2P and alternative, undesirable traffic payload as opposed to source and destination ports. Such approach poses a large overhead and presents a delay if every packet's payload are examined. There are commercial products available that address the above issues, but those might be expensive and inflexible for a campus community. D. Ennis et al. [8] proposed a model which combines a several open-source solutions based on a policy enforcement framework. The key component of such network policy enforcement is the signature-based Intrusion Detection System (IDS), which uses the principle of attack signatures to identify attack or undesirable traffic on the network. This approach has been extended to detect traffic that violate the organizational policy. Upon policy infringement, IDS communicates with a policy enforcement plug-in that is capable of limiting access to the offending computer system (e.g. blocking traffic from source). The policy enforcement plug-in in their implementation was SnortSam that resides in the firewall (IPTables) system, and it modifies the firewall rule-set based on instructions received from the IDS. The plug-in also sends log messages for the changes made in the firewall rule-set to a centralized log server (Syslog-ng). The centralized log server has a log parsing utility as depicted in Figure 3. Hence, with a centralized log server in a place and a log parsing utility, the administration can access the details of logs and other historical information from a web based front end.

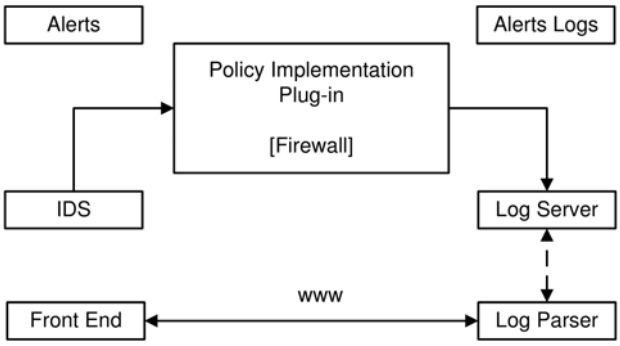

Fig 3. The final Model Policy Enforcement Framework [8]

The approach of D. Ennis et al. [8] is very realistic, simple, and does not require a great level of expertise, but the network infrastructure has to handle and process the log messages the pushed towards the centralized log server.

3.5 VoIP Spamming Attack (Vamming)

Voice Over IP (VoIP) spamming is one of known attacks that has quickly became an issue with the ever-rising popularity of VoIP. Such spam calls (Also named vamming) raises out the authentication limitations in the Session Initiation Protocol (SIP) within the P2P domain. With the Internet as it is a habitat nature of the VoIP environment, it also provides a good soil for vamming. The openness of the IP-based network like Internet allows anyone to join the network without necessarily presenting authentic identify that can be trusted. While in managed networks the problem of user trustworthiness could be controlled, it is a gruesome issue in distributed, un-managed P2P networks. One of the possible approaches to solve the issue of user trustworthiness in P2P networks would be associating some level of trust with a specific P2P entity with using one of the well-known network security protocols (i.e., TLS or SSH), but with a participatory and large ever growing P2P environment such an approach would not be easily achievable in un-managed P2P networks.

Another approach introduced in the study of N. Banerjee et al. [10], where they suggested a possible solution to the user trustworthiness issue in P2P networks. They proposed a trust enforcement framework consists of computation and memory bound functions that associate trust implicitly to the P2P VoIP entities. Based on the associated trust factors, one can judiciously decide whether a call from P2P VoIP entity can be accepted or not, which forms a preliminary screening mechanism against malicious callers. Such user trustworthiness framework could also be applied to other P2P network entities such as download and share file engines and applications.

The growing popularity of IP-based telephony systems is reflecting the increasing demand for VoIP P2P services. While currently most implementations focused on proprietary signaling protocols, in the future, they will all be based on the SIP [10] for compatibility reasons. The SIP is one of the most robust VoIP communication protocols in packet-based networks, as it truly fulfills the full potential of P2P networking by supplying the capacity of interoperability between different VoIP clients. With the rising attempts to integrate SIP into P2P network domain, such systems could suffer from inherent P2P security issues such as trust, privacy, authentication etc. While Some of these issues can be solved by applying end-to-end encryption techniques, the user trust and identity management issues remain as major problem in P2P networks.

Given that the architecture of SIP works on an end-to-end basis without any network infrastructure assistance, the connection can be formed directly between the two endpoints. The SIP can offer either network-based VoIP service or P2P VoIP-based service to the VoIP

network service providers. Nevertheless, the user registration offered by SIP does not guarantee user trustworthiness as it widely used within free online facilities, which allows users to easily create additional accounts in the networks of service providers. Therefore, it is targeted by malicious users, who aim to exploit the service to launch a voice spamming attack. Note that SIP architecture includes a number of security features that enables user agent authentication, message confidentiality and integrity, and hiding personal information. However, SIP can face security challenges when deployed in P2P networks. The following discussion describes SIP's security mechanisms and the reason why they could fail to satisfy various conditions in the P2P domain. The SIP authentication is provided by using HTTP digest authentication. Such a digest authentication operates on a mutual confidential basis (e.g., shared password), but not all communicating entities in P2P networks are completely unknown to each other, which makes the above authentication option not feasible (i.e., it would require the communicating entities to set up a call to share a secret). On the other hand, secure SIP signaling could be transported over authenticated transportation layer utilizing TLS certificates issued by mutually trusted certification authority. The trusted third-party in large scale pure P2P networks is not possible. SIP also can offer email message encryption integrity and utilizing public key encryption scheme. Again, a third party certification authority is involved to issue and authenticate public/private key pair [11]. Hence, the vamming issue in VoIP P2P networks comes from the limitations of SIP security features to adequately authenticate peers in P2P networks. Therefore, an authentication mechanism to check the caller's identity becomes essential in pure P2P networks. Once the identity of a peer is established a reasonable amount of trust can be associated with the caller's identity.

As highlighted above the standard public key encryption alone is not sufficient for P2P network to guarantee the ownership of the public keys used by communicating parties when such keys are not ensured by collaborating certification authorities. Therefore, to overcome vamming attacks, there should be an authentication mechanism to verify the caller identity even if the caller is unknown to the callee, which is not easy to achieve in the absence of an identity federation between the caller and the callee. Thus, the proposed solution in [7] to the authentication problem in P2P is to use public key encryption with one-way hash function to evaluate the trustworthiness of an unknown caller. Hence, the authors have introduced a new parameter to the authentication process, called token, as a part of the caller identity in addition to the public key. In the proposed authentication scheme [10], the caller's identity is generated based on the following relation:

$$last(H(token), b) - last(publickey, b) \quad (1)$$

Where H is the one-way hash function, and b is a value < the length of $H(token)$, the length of caller's public key, and the $last(p, q)$; whereas the p and q are the caller's prime numbers. The b must returns the last q bits of the strings p.

Note that the above proposed scheme can provide $2^{(m-b)}$ distinct tokens for each public key. The strength of the caller's identity C_{id} (i.e., to be trusted by other parties) increases along with the increase of b value as follows: $C_{id} = \frac{b}{(m-b)}$, where m is the length of $H(token)$. When C_{id} is made large enough as a system requirement, then it will be computability expensive for the caller to start vamming, especially when the used hash function is designed to be computationally harder with the increase of b value [10]. As the authors stated, the relation behind the term of identity strength is that only genuine callers would spend considerable time and CPU resources under such an authentication process to generate authentic identities.

When the caller makes a new call, his identity strength (i.e., user trustworthiness as stated by the authors) will be evaluated against a specified threshold T as follows: $T < C_{id}$, in which the call will be accepted; otherwise the call will be rejected if the caller identity is not strong enough to be trusted, and it could be part of the vamming attack. As the threshold T plays critical role during the caller identity evaluation, the selection of the T depends on many factors as discussed below. First, the selected hash function in the proposed authentication mechanism should be strong enough according to the today computing economics and computing machines' capability, maximum allowable number of identities for each caller, and $T < b$. To overcome an external attack such as a man-in-the-middle attack (MITM), the above P2P authentication protocol specifies that the caller have to encrypt the generated token with his private key, which can be decrypted by the callee using the caller's corresponding public key. Also, the authors highlighted the risk associated with distributed generation of public key, in which two different entities could generate the same public key. However, such risks can be reduced if the chosen p and q primes numbers for public key generation were very large, especially when the public key is used for real-time VoIP calls only. However, their protocol did not discuss how the callee can verify the authenticity of the caller's public key. Note that the token and public key based identity generation mechanism discussed above does not limit the capability of a single caller from generating simultaneously unlimited number of valid identities. Therefore, the authors have leveraged the public key generation with memory-bound function described in [12] as a constraint mechanism to limit the capability of a rogue caller against generating unlimited identities to initiate

vamming attack. They utilized CPU/memory bandwidth as bound function; instead of CPU bandwidth only, because CPU/memory bandwidth varies greatly across computing machinery, compared to CPU bandwidth only.

For public key generation, the authors discussed two scenarios. One, where a trusted third party (named service provider) is available, and the second scenario, where there is no such trusted third party involved in the process. Although the first scenario is widely discussed in the literature of public key cryptography and key management, the public key generation scheme in [12] introduced a memory-bound function for the public key generation process to restrict the capability of uses to obtain unlimited number of public keys. In the second case, where there is no service provider, the authors suggested the use of non-interactive version of memory-bound function based verification specified in [12], which delegates the key generation task completely to the user. In such case, the caller has to provide a list of cryptographic parameters specified in [12] to the callee as part of the call processing protocol, including the specified *F()* memory-bound function. These cryptographic parameters are provided to the callee along with the caller's identity tuple of the token and the caller's public key. Moreover, the caller has to build and send a tree of pre-images of cryptographic parameters as presented in [10], [12]. Hence, the limitation of this procedure is that the length of the message(s) exchanged during call establishment can be very large.

3.6 Trust Models for P2P Network Security

Many trust models for P2P network security have been proposed to detect malicious peers in the networks. Fan et al. [13] proposed behavior-based trust model that calculates the trust rating and recommendation matrices for the P2P file-sharing networks. Measuring the trustworthiness of peers accurately in not a trivial task in pure P2P networks, and conventionally it requires collecting the historical behavior of each participating peer in the network to evaluate their reputation. Capturing peers' transaction behavior in P2P networks appropriately can be challenging since each peer in the P2P network can take different roles (i.e., client, server, or both). Moreover, it requires the other peers in the network to provide trust recommendation value to the network community regarding the transaction behavior of the serving peer. In their paper, the P2P network is presented as a directed weighted graph *G(V, E)*, where *V* represents set of peers, and $E = \{i \mid j \in Trans\ (i), l_{ij}\}$ denotes the resulted personal trust rating after a transaction. The *Trans(i)* represents set of peers form which *i* has received service, and l_{ij} denotes the trust rating from peer *i* to peer *j*. They defined personal trust rating l_{ij} as the percentage of successful transactions that *i* has satisfyingly received form *j*:

$$l_{ij} = \begin{cases} \frac{\max(S_{ij},0)}{\sum_k \max(S_{ik},0)} & if\ \sum_k \max(S_{ik},0) \neq 0, \\ 0 & otherwise \end{cases} \quad (2)$$

$$S_{ij} = successful(i,j) - unsuccessful(i,j) \quad (3)$$

To set the landscape for their proposed solution, the authors of the above mentioned paper clearly defined the difference between the peer trustworthiness and reputation notions in the P2P network context, which is adopted from earlier work in [14], [15]. They defined peer reputation as collective trustworthiness measure based on trust ratings that receive from other members in the network, while the peer's trustworthiness can be a local value (e.g., personal opinion or score) in the other peer that was serviced by the aforementioned trusted peer. Therefore, in their proposed trust model, named Dual-EigenRep, they take into account the resource service behavior and the trust recommending behavior; hence, they introduced two interrelated reputation measures, RDRV and RGRV, for evaluating the reputation and the trustworthiness of the peers in the P2P file-sharing networks. The recommended reputation value (RDRV) for each peer aggregates the recommending reputation values (RGRV) of other peers that have received service from this peer in order to evaluate the resource service behavior in the view of other peers. The recommending reputation value (RGRV) for each peer aggregates the recommended reputation value (RDRV) of other peers from which this peer has received services, including itself trust ratings in order to evaluate its trust recommending behavior. These reputation values tightly depend on each other, and they can form a mutual reinforcement procedure for the reputation values among the peers in the network, which subsequently can spot the transaction behaviors of malicious peers as stated by the authors. However, detecting malicious peers is delayed until the convergence process (i.e., dissemination of the collected reputation values among peers) is completed over the entire network.

This critical point is also highlighted in Kun et al. [16] research, as they stated that reputation-based trust models rely on the previous interactions of the peers, and the collection of local trust values, which requires more time to gather such information across all the peers in the network. Besides, the computational complexity is increased due to the above collection process of the reputation values. While the author classified P2P trust models into the following four categories: Identity-based trust model, role-based trust model, reputation-based trust model, and automated trust negotiation model, they stated that most existing trust models in the literature is based on reputation methods [16]. Xiao-Yan [17] proposed a punish mechanism as an integral part of his trust model for P2P networks to encourage nodes to share resources and trade fairly. Therefore, the proposed mechanism divides nodes

in the network into four groups based on their assigned trustworthiness values according to the nodes transaction behavior and the quality of service. The nodes trustworthiness value can be changed according to its interaction with other nodes, and the selfish node that requesting services but does not serve others will be punished.

4. Some Of Popular P2P Implementations

4.1 SKYPE (Super Node Concept)

The study by Baset and Schulzrinne [18] highlighted that many of the P2P implementations have used the concept of a Super Node (SN). A SN is a selected peer which has a public IP address and could be a specially seeded with the distributed program. SN's allow us to transit Network Address Translation (NAT) boundaries as well as firewalls by acting as a proxy for the IP session. This extra routing node does not impact call quality in the well-designed Voice Over IP (VOIP) protocols, and it can facilitate the sharing of calls between users via 3-way calling [18]. After the analysis of Skype protocol has been validated, Skype claims that the Skype protocol, using SNs, can find a user of the protocol if the user has logged into the network in the last 72 hours. However, there are a few drawbacks to the SN concept leading to key weaknesses. The SN's are still bound to a centralized log-in server. The Skype authentication protocol is still handled by a centralized login server. This server is a weakest link in the survivability of such networks, as its demise would cause a Denial of Service on the network and should be considered for P2P networks. Trust of the authentication server and the peer nodes must be guaranteed on a network where repudiation is a requirement. Since the Skype protocol is encrypted with AES, Baset and Schulzrinn were not able to determine if the nodes validated the sender's identity. This lack of trust and affirmative authentication could impact the adoption of such P2P networks, especially in telecommunications and business applications. It is understood that a log-in server may not be necessary in all P2P applications.

Another interesting problem raised by Baset and Schulzrinne work is that the P2P protocol in Skype takes the role of super node in proxying information for end point nodes behind a Firewall or NAT device. While the Skype transaction is encrypted using AES, due to Skypes closed format, it is not known how the key exchange is handled, and thus the threat of a man-in-the-middle attack still there. While Skypes protocol should be commended for its inclusion of robust encryption technology, its lack of transparency and open design along with the centralized log-in leave some weaknesses regarding its security.

4.2 Internet Telephone and SIP

Singh and Schulzrinne [19] proposed and implemented a P2P Internet telephony framework using Session Initiation Protocol (SIP). The framework supports various advance services, but it lacks various security related features, which the authors discussed for future study. The paper focused on the secure routing, however; privacy, confidentiality, and DoS attacks still major threats to their architecture. Additionally, the authors mentioned a malicious DHT node might not always accurately relay call requests or record any possible misconduct call requests. In their design of hop-by-hop routing of requests and replies, each hop (peer) updates the source identifier for confidentiality purposes [19].

The authors also proposed and implemented a P2P Internet telephony framework using Session Initiation Protocol (SIP). The framework supports various advance services, but it lacks various security related features, which the authors discussed for future study. The paper focused on the secure routing, however; privacy, confidentiality, and DoS attacks still major threats to their architecture. Additionally, the authors mentioned a malicious DHT node might not always accurately relay call requests or record any possible misconduct call requests. In their design of hop-by-hop routing of requests and replies, each hop (peer) updates the source identifier for confidentiality purposes [19].

4.3 P2P-Based NFS

Kosha is a P2P enhancement to Network File System (NFS) proposed in the work of Butt et al [20]. By leveraging the P2P to use the wasted client resources, organizations could meet some of their NFS needs. The authors show how Kosha could be used securely to decentralize file storage using an NFS-based model that utilizes the slack client space for storage. This work presents the variety of system operations that P2P networks can support.

4.4 KaZaA (Usability and Privacy concern)

In P2P file sharing system, one of the major concerns is that users may be sharing private and personal information without their knowledge. Although P2P file sharing systems such as KaZaA, Gnutella, Freenet are primarily intended for sharing multimedia files, but they might be exploited and become an access point to users private and personal information. The study by Good and Krekelberg [21] highlighted a case where a user of KaZaA file sharing who interfaced in their work and became participant in unacknowledged file sharing. Here are two major issues that users are facing: Inexperienced users unable to tell which files they are sharing, and sometimes incorrectly

assume they were not sharing any files, while in fact they share many of the files they have downloaded previously or other files in their hard drive. Analysis of KaZaA has shown that a large number of users were unaware of the fact they were sharing sensitive and private files, which consequently exploited by other users, who download the incautious user users' files, which contain personal information.

The attraction of P2P downloading ease and its file sharing features motivate some users to utilize the service without getting sufficient experience, which could lead to unguarded configurations of service's application. The main approach to prevent the problem of unacknowledged file sharing is the proper set up and use of the applications user interface. To observe if indeed some users were downloading other users' private files, an experiment was conducted and designed in such a way that a dummy client was designated to run with populated dummy files. Among dummy files were CreditCards.xls, Inbox.dbx, Outlook.pst and other type of files that were intended to appear to be private [21]. The dummy client was set to run for 24 hours period. From a dummy server, a total of four unique users have requested to download spreadsheet named CreditCards.xls, and four download requests from two unique users for Inbox.dbx files. This experiment has suggested that the system abuses are occurring, and the frequency of such events is considerable [21].

Based on a list of security and usability guidelines pro-vided by Good and Krekelberg [21] that can be adapted for Peer-to-Peer File sharing applications. This list takes into account the unique demands of continuously connected systems that distribute personal files. Any P2P file sharing application can be safe if users:

1) Are aware of the files that are being downloaded from their machine.
2) Have the complete control and sufficient experience to share or stop sharing their files.
3) Avoiding taking risks that could lead to misconfiguration of sharing or exposing private files.
4) Are aware and comfortable that their systems are handling the process of file sharing correctly.

Also, Good and Krekelberg [21] conducted a cognitive walkthrough of end-user case study for KaZaA file sharing system, while paying a close attention to whether or not the KaZaAs interface was able to meet the above guidelines, and if not, why were users confused.

4.5 P2P Intrusion Detection System Based on Mobile Agents

Traditional Intrusion Detection Systems (IDS) are based on the idea of centralized coordinator within a hierarchical architecture. The work by Ramachandran and Hart [22] proposed a different approach; a peer-to-peer intrusion detection that has no centralized coordinator. The peer-to-peer intrusion detection approach can be compared to a neighborhood watch, where the neighbors look out for each other when a virtual neighborhood is originated. If an intrusion is detected, it is then reported to the whole neighborhood and the neighborhood takes a collective action. In addition, mobile agents are used to cooperate in the detection process. Each site probes its neighbor periodically using mobile agent to visit and check up on it and report back. When an anomaly of intrusion is observed, the observer-neighbor initiates a voting process to take countermeasures against the suspected site [22].

As the proposed distributed mobile-agent based intrusion detection system has no centralized coordinator, it can scale well and be updated accordingly along with the network size. Each neighbor in the neighborhood has information related to the safety of a site, but there is no inherited trust among neighbors. The neighbors use the distributed data store located on each other to ensure the integrity (i.e., no tampering of files) and the site is functioning correctly. Mobile agents are then used to do intrusion detection analysis, which improves the intrusion detection services as it makes them more responsive and dynamic. Such a mobile agent based IDS can be easily customized to fit specific needs of large networking environments [22].

Their proposed architecture consists of Chief, Detective and Cops [22]. Cops are mobile agents, and have many different tasks and can be dispatched to different sites. Different cops may perform: check-sum of sensitive data files, and system applications; look for any change on access and location of sensitive fi ofles; verify the consistency of log files; look for signatures of known viruses/worms and; watch the CPU load of a site; analyze the activities recorded in the logs for networking events. Cops execute at the site and report results to the detective. The detectives' job is to dispatch cops to different sites and various neighborhoods and supervise cops. The detective observes and analyzes all the Cop's reports if a suspicious activity is detected then the detective informs the Chief. When Chief receives reports from detectives, it should decide whether there has been an intrusion in the neighborhood or not. If there is an intrusion, the chief then sends a voting call to initiate measures against this neighbor. This is accomplished by delegating a voting agent to a randomly chosen neighbor of the suspected site. Note, in each site, the Chief is responsible for participating in such voting process. In this case, the Chief authenticates the digital signature on the received voting call, and then it assesses its observation regarding the suspected site before it forwards this voting sheet on to the other neighbors of the suspicious site [29]. The advantages of the proposed IDS as follows:

1) Flexibility: new cops, detectives and agents can be added without disturbing the neighborhood.
2) Distributed decision making.
3) Scalability: neighborhoods scan well for large network by simply keeping neighborhoods small and creating new neighborhoods
4) No inherent trust.
5) No single point of failure.

This IDS design for P2P networks aims to enhance the performance of the current available IDS systems. It can provide additional protection for network resources as well as it can be integrated with other network security appliances such as virus detection tools and firewalls. Since the above IDS is based on a peer-to-peer distributed model, it forms a robust IDS structure and also makes it harder for someone to compromise data and impact the whole system [22].

5. Reputation, Trust, and Secure Routing Architecture in P2P

5.1 P2P Security Layer (P2PSL)

There have been attempts to address security in P2P, and P2P nodes, for example by adding a flexible security layers. The study by Detsh et al. [23] lay out a design for P2P Security Layer (P2PSL). By adding a security layer to the architecture we can add much more flexibility to the P2P protocol. Note, not all nodes may need the same security requirements. One node may require confidentiality; another could require non-repudiation or authenticity. This security layer contains the need security functionality into P2P applications [23]. The authors focused in designing the P2PSL as a framework, which is implemented and configured independently from the application. They outlined in their design the separation of the security layer from the communication layers. Some of the benefits to consider include: the ability to tailor the level of security for the application, the ability to slowly scale up the security of the protocol using a common API, common standards for end users, greater interoperability between P2P applications and protocols. P2PSL allows nodes to have authentication, confidentiality, logging, and compatibility with other nodes that do not have such security functions. The flexibility in the design allows this generic implementation to be used across other platforms and be backwards compatible with nodes not implementing security layers. This backwards compatibility can be useful in incremental network upgrades.

P2PSL was written in Java and JXTA/JAL. There is a configuration module that is used by P2PS and is stored as an XML file. The application then uses JAL to interface with the module adding whatever requirements the user perform with the configuration module. The P2PSL adds the necessary cryptographic components to any secure design for P2P systems but is still limited by either trust relationship among peers or the need for a common log-in server, which raises concerns in a decentralized networks.

5.2 Secure Mobile P2P

Other investigators are also interested in a common security layer and are working to standardize them. Walkerdine and Lock in [24] investigate a mobile P2P security architecture and have shown the benefits of the common security layer model. They specifically analyzed PEPERS; a solution to the mobile security problem. Mobile security introduces many unique problems in the P2P context. Authentication for remote users as well as security updates and confidentiality have become more difficult in a mobile environments. The ability of 3/4G wireless, traditional 802.11 wireless, and bluetooth networks to serve as a transport layer for P2P applications makes security ever more important and complicated. Wireless networks add another threat dimension of snooping, which is partially mitigated with the traditional cabled networks. Walkerdine and Lock proposal was based on their previous work in P2P security with PEPERS scheme. Their proposed architecture uses a modular security layer for encryption, authentication, trust, and etc. [24]. As for security and trust, Walkerdine and Lock analyzed work in the field employing a distributed PKI system to ensure confidentiality, integrity and authenticity. PEPERS seeks to work with some of the constraints placed upon mobile networks, such as: communications cost, battery life and capabilities ¥cite{14}. The key objective is to having a generic layers to allow heterogeneous platforms, operating systems and protocols to work together to ensure security. In fact, PEPERS has a layered design as outlined in below [24], [25]:

- Operating System (OS) abstraction layer: This layer attempts to serve as a abstract layer between the OS and the PEPERS platform. By abstracting out the OS the goal is to maximize the number of platforms able to run the platform. This allows the relevant API calls to be made from the platform into the OS.
- P2P network layer: Provides a generic abstraction to a variety of P2P protocols to allow communication in a mobile environment.
- Security layer: Since PEPERS is designed for security, it has been broken into sub modules. These modules are expandability. The ability to add new functionality grant the customizability to the end- user to use or not use any or all of the security functions. Interoperability, the ability for application to leverage or replace PEPERS

modules. Modularity, as we have been discussing a modular design.

- Security management module: A module to administer the system security setting and store them.
- Device information lookup module: The ability of the system to query information about the platform it is running on.
- Platform management module: The module that controls the system.
- P2P communication module: Handles P2P networking functions such as publication, discovery, communications and general P2P management.
- Data repository module: The management module for any stored data, either the hosts or peers.
- Peer recovery module: Allows peers to roll back transactions and recover lost data.
- Trust module: A method to provide a reputation system in the PEPERS platform.
- Logging module: In general, best security practices have accountability and logging as a foundation.
- Dynamic verification framework: a module that ensures the security of loaded applications.
- Authentication and authorization module: Authenticates both users and devices and ensures access controls are enforced ¥cite{15}.
- Encryption module: Provides encryption functionality for the modules, including authentication as well as confidentiality.
- Data lifetime module: Peer data may have a shelf life due to its sensitive nature.

5.3 Reputation, Trust, and Security

Dewan and Dasgupta mentioned that a reputation based systems can, with certain assumptions, considerably reduces the number of malicious transactions in decentralized networks [26]. They enumerated the attributes necessary for a reputation system. The first of these is the owner or provider of the service. This is the person or system providing a service on the network, and reputation is always associate with an entity. The next attribute is the context, service or attribute. This is the service being offered by the entity. The reputation is based upon an evaluation of the service rendered. The requestor or evaluator is another necessary part of the reputation system. A requestor makes demands for services offered by the owner or provider. The transaction is the result of the demand by the requestor and the service rendered by the provider. This should be reduced to an atomic value so tracking and metrics of the transaction can adequately track it. The recommendation for the reputation may be either quantitative, subjective or possibly a hybrid of the two. This is the basis for scoring the transaction, and it used by the metrics to aggregate the reputation. Metrics and accumulators then take the recommendations and score them for their reputation. This may take into account the probability of a future transaction being trustworthy. Finally, the storage of the reputation must be secure so that it is not tampered with.

In ad-hoc networks the rate of good-put goes up with use of a trust system on a P2P network. However, identity authentication remains a fundamental problem in any trust system and could be mitigated by using digital certificates. Combining trust and identity with the schemes presented in [23], [24] could provide a secure platform for reputation and identity to function; however, clearly without cryptographically strong schemes in place the issue of identity authentication still presents challenges in such an environment. Reputation systems also must choose the correct metrics of trust to ensure accurate results.

Yu, Singh, and Sycara in ¥cite [27] examined a method of trust using PKI to verify identity with a binary trust system, which uses referrals through which peers help one another find witnesses. The authors focused on systems which share files and thus makes rating fairly easy. Binary ratings work pretty well for file sharing systems where a file is either the definitive correct version or is wrong, but cannot accurately model richer services in other settings such as web services and electronic commerce, where a boolean may not adequately represent a peers experience of the quality of service (QoS) with other peers, e.g., the quality of products the peer sends [27]. This analysis limits the scope of [27] work to a subset of peer networks, but their work does lay a solid foundation for a trust metric.

The work by Damiai el al. [28] showed that standard trust models from wired networks may not be effective in wireless P2P networks. Many wireless P2P network lack the resources to use traditional cryptographic schemes and may rely only on trust methods. However, they pointed out many of these trust systems default to trusting nodes, but there are chances that P2P users on mobile encounter strangers. In addition to trust issues, using a PKI solution would be extremely difficult and to some extend insecure since a naive delegation and replication of the CAs responsibilities makes the service more vulnerable [28]. If centralized authentication is impractical or insecure, and trust is: (1) extremely vulnerable to dishonest actions, and (2) potentially very inaccurate due to the absence of fixed trust infrastructure and the ephemeral nature of the connections [28].

Trust can be very difficult to poll in P2P systems to weed out malicious nodes. The work in [29], Damiai el al. lay out a way for nodes' neighbors to query the reputation of a node. One of the fundamental problems of trust in a

P2P network is the issues of identity authentication, and to remedy this, not by digital certificates, but through use of sustained identity. That is, the reputation is tied to the length of time a host on the P2P network remains with the same identity. This aims to keep malicious nodes from continuously reacquiring a new identity in the P2P network and exploiting trust. This approach has merit in the realm of mobile P2P (M-P2P) networks where network connections to nodes are only transitory. By polling nodes who have had access to this node reputation voting could aid in the identification of safe nodes for both routing and for collaboration.

5.4 Cost Leads to Security

The investigation by Rice [30] presents an interesting option for reputation and security in P2P networks. He proposed the use of the Pearson Coefficient to charge nodes higher cost when connecting to malicious nodes in the P2P networks. Thus, using incentives to choose connections that increase the network's resilience against the propagation of malicious code [30]. Using incentives may be a strong catalyst to security evolution in P2P networks and its use could help to weaken malicious nodes.

5.5 P2P Admission Control

Most works in P2P security concentrated on authentication, key management and secure communication. However, secure admission issue remains largely untouched and unexplored. Secure admission issue is an important prerequisite for many P2P security services; otherwise how one becomes a peer in P2P settings. Saxena et al. [31] presented their work that introduced a peer group admission control framework, which is based on various security policies and cryptographic techniques. In their work, they assessed the effectiveness of concrete P2P admission mechanisms over various cryptographic techniques. Their analysis focused primarily on performance, but other important features such as anonymity, unlikability and accountability were included.

However, the process of how peers become part of P2P networking paradigm is not fully covered. There are currently many operating P2P networks that either operate in a completely open access (i.e., no admission control in place) or they admit peers based on some ad hoc methods.

While the secure communications are based on key management, trust management and access control, all of the above are actually effective only after a member is authenticated to be allowed to join the group. The argument is that without a secure admission process there is no point in deploying other security measures such a secure key management and trust management since a malicious prospective member can easily falsify his identity and join the network. The admission classification can include simple admission control functionality, such as static ACL-based admission, or admission based on the decision processed by fixed entity; either external (e.g., TTP) or internal (e.g., a group founder). An in-depth discussion of various issues regarding admission control policies can be found in [32]. When peers start to be responsible for admitting new members themselves, then the admission policy become a form of limited consensus among related peers in the network. This limited consensus method is essentially forming an agreement threshold of current members who agree to allow the prospective peer to join the group. Hence, two thresholds are specified: fixed and dynamic. The distinction between a fixed threshold and dynamic threshold is that a fixed threshold specifies the minimum number of votes to allow the prospective to join, whereas a dynamic threshold specifies a fraction of the current group size to take the decision.

Group sizing and population can be problematic in P2P environment since peers in P2P networks join and leave the group at any time. Therefore, for dynamic thresholds, determining the group size is the base step in the process. In asynchronous P2P setting, the straightforward way to deal with the group problem is to impose a trusted authority that is updated constantly with the membership information, such authority is referred to as Group Authority. However, this Group Authority can also be a single point of failure if exploited.

The study in [31] have implemented a group admission control toolkit that consists of cryptographic functions that were developed using Open SSL Library, the toolkit is written in C programing language over Linux system. They further experimented and integrated the toolkit admission control with Gnutella and Secure Spread to evaluate the performance in the context of real P2P systems with setting up dynamic and fixed thresholds. The experiments conducted by the authors demonstrated that unfortunately, advanced cryptographic methods such as verifiable threshold signatures are not sufficient for the current P2P applications.

5.6 P2P Location Management with Trust

An interesting take on trust could be derived from location awareness of peer nodes. The location of a peer could impact the level of trust if the peer was located in a known or expected location.

When a mobile host moves to the new network it gets assigned a new IP address for routing purposes. For the nodes that wish to communicate with the newly relocated mobile host, the new IP address needs to be known and disseminated. This new IP address of mobile host must be stored at specific networks servers. A static configuration for mobile host address leads to poor performance,

especially in distributed architecture of P2P solutions as it presents security and mobility issues.

Sethom et al. [33] proposed a Secure P2P Architecture for Location Management. (S-PALMA) in which the mobile node's current address is stored at relatively stable locations near to the place where it can be reached. The S-PALMA architecture is mainly based on tapestry location and routing algorithm providing several desirable properties: Flexibility, scalability and simple fault handling. In tapestry location with routing algorithm, each node maintains a routing table consisting of node IDs and IP addresses of the nodes with which it communicates. The tapestry then uses these local routing maps at each node to route overlay messages to the destination node.

The P2P network are subject to specific security threats such as denial of service, which is probably the major threat on P2P service discovery. Another types of attack include: unauthorized use of services (access control is needed to prevent this attack), man-in-the-middle attack and identity usurpation (allows attackers to control service discovery and lead to eavesdropping). Since location management could represent as a discovery mechanism, any attack on it essentially an attack on the discovery service. Some of the common techniques of such an attack include: incorrect lookup routing by forwarding look-ups of non-existent nodes, sending erroneous routing update messages or partition attack that allow a group of colluding nodes to hides the real one in order to cause a DoS or get valuable information about legitimate networks behavior. The S-PALMA architecture presents a layered security mechanism that collectively protect the location services, and provides security access control, message authenticity, confidentially, and anti-replay [33].

The S-PALMA architecture composed of a group of Distributed Lookup Servers (DLS) that are interconnected via heterogeneous networks. These DLSs form an overlay network that resolve queries from Client Nodes (CNs) while also publishing location information among servers in the overlay network. The forementioned interconnections in the overlay network are maintained using tapestry node insertion and neighbor notification algorithms. Each DLS is assigned a new identifier (ID), which is generated using hash function (SHA-1) with the DLS' IP addresses as input for the hash function. Nodes either Mobile Servers (MS) or clients at a given time, and they communicate their presence over DLSs, which also process queries received from them. When a network node send a query to one of the DLS, it will be served from that particular DLS. The message confidentiality mechanism in S-PALMA is designed to accommodate the mobility constraints since public-key operations could present considerable load [33]. The S-PALMA architecture has also some other advantageous security properties namely: malicious updates by unauthorized hosts are prevented.

5.7 Public Key Infrastructure (PKI)

P2P sharing application is very popular in daily basis communications. However, due to security concerns with respect to confidentiality and integrity, these applications are not gaining traction in the corporate world. Berket et al. [34] proposed a PKI based security solution for the P2P applications so that the corporate world can use P2P with confidence. Unfortunately, the forementioned solution has some limitations, and the door is left open for further research in the area of confidentiality and integrity issues with respect to routing, as it is difficult for group members to have a common, shared secret key for protecting their communications while achieving dynamic grouping. Therefore, a newly generated key is needed when the membership status changed to protect forwarding secrecy in the network. In [34], the authors use distributed, mechanisms for re-keying called SGL. However, SGL does not provide the details of the mechanism specifications. They developed a group policy and associated mechanisms to guarantee that every peer in a group proceeds with the same policy. In such system, peers can build their trust relationships autonomously. An underlying associative trust still exists since peers essentially become owners of data they download [34]. The framework allows users immediate access to the application. In the their work, the authors present a PKI-based security mechanisms which can be used to provide PKI-based security for peer to-peer information sharing.

5.8 Multi-Path Key Exchange

The key exchange is difficult in P2P systems with SNs, especially in a pure P2P system with Distributed Hash Tables (DHT). In a pure P2P network with a DHT all nodes are equal, and the authentication in such system without a centralized server complicates the process. The work by Takano et al. in [35] have analyzed this problem and have proposed a method for the key exchange that relies on direction and probability. While this method will not protect a pure P2P network at this time, its work is promising and warrants further investigation.

5.9 Secure Routing in P2P Overlay Networks

The P2P nodes may utilize overlay network in order to communicate between them. Malicious nodes can prevent secure communication between two peers with respect to message delivery. The study by Castro el al. [36] presents various methods to deter these attacks. The work states that the proposed methods can allow nodes to join the overlay network with maintaining reliable routing state and massage forwarding security.

In a structured P2P overlay networks, there is an existing notion that peer nodes are non-malicious in nature. However, this assumption is false. A malicious node can attack the P2P overlay network. In [36], the authors discussed the design of techniques for secure node joining, routing table maintenance, and message forwarding in structured P2P overlays [36]. The secure routing techniques can be integrated with existing techniques to construct robust and secure applications. These techniques allow us to tolerate up to 25% malicious nodes while providing good performance when the fraction of compromised nodes is small [36].

6. Conclusion and Future Work

The need for a standardized approach to secure P2P networks becomes more clear as we study the structure and growth of the networks. If P2P networks can implement a common security layer(s) that all nodes can then share, then it can be utilized as a vantage point for to customize the P2P network security based on need. In addition, we have seen how the super node concept can aid in network security by offloading some bootstrap functionality to keep a stable network. Hence, this could also be utilized through building security operations into the core super node functionality. Also, our work highlighted that trust remains vital to P2P networks and is shown to be a very difficult problem due to nature of P2Ps systems. Malicious nodes either masquerading or misusing the network have presented their own set of problems from classic man-in-the-middle attacks to the propagation of worms or altered contents. Therefore, a comprehensive solution that can tackle all the security issue mentioned above has become an essentially for secure P2P networks.

As we were conducting our research, we have noticed a repeating theme. Virtually, every secure P2P model we have examined came down to a fundamental principle of trust for P2P security. However, three is a lock of comprehensive trust model with secure routing foundation for P2P nodes. Any intended security should also take into considerations network latency impacts when the solution is deployed. For the future work, we plan to analyze the effects of adding cryptographic techniques to the routing nodes in P2P networks. This will enable trust in a manner similar to PGPs web of trust. Furthermore, we also intent to model a scenarios with different attacks on security of P2P systems and measure the results when no security in P2P is implemented and when security solutions are set in.

References

[1] C. Selvaraj and S. Anand, “A survey on security issues of repu-tation management systems for peer-to-peer networks,” Computer Science Review, vol. 6, no. 4, pp. 145–160, 2012.

[2] D. S. Wallach, “A survey of peer-to-peer security issues,” in Software SecurityTheories and Systems. Springer, 2003, pp. 42–57.

[3] L. Washbourne, “A survey of p2p network security,” arXiv preprint arXiv:1504.01358, 2015.

[4] D. K. Bhattacharyya and J. K. Kalita, DDoS attacks: evolution, detection, prevention, reaction, and tolerance. CRC Press, 2016.

[5] N. Naoumov and K. Ross, “Exploiting p2p systems for ddos attacks,” in Proceedings of the 1st international conference on Scalable information systems. ACM, 2006, p. 47.

[6] P. Maniatis, T. J. Giuli, M. Roussopoulos, D. S. Rosenthal, and M. Baker, “Impeding attrition attacks in p2p systems,” in Pro-ceedings of the 11th workshop on ACM SIGOPS European workshop. ACM, 2004, p. 12.

[7] G. Chen and R. S. Gray, “Simulating non-scanning worms on peer-to-peer networks,” in Proceedings of the 1st international conference on Scalable information systems. ACM, 2006, p. 29.

[8] D. Ennis, D. Anchan, and M. Pegah, “The front line battle against p2p,” in Proceedings of the 32Nd Annual ACM SIGUCCS Conference on User Services, ser. SIGUCCS ’04. New York, NY, USA: ACM, 2004, pp. 101–106. [Online]. Available: http://doi.acm.org/10.1145/1027802.1027828

[9] L. Zhong, M. Kihl, and X. Wang, “Topological model and analysis of the p2p bittorrent protocol,” International Journal of System Control and Information Processing, vol. 1, no. 1, pp. 54–70, 2012.

[10] N. Banerjee, S. Saklikar, and S. Saha, “Anti-vamming trust en-forcement in peer-to-peer voip networks,” in Proceedings of the 2006 international conference on Wireless communications and mobile computing. ACM, 2006, pp. 201–206.

[11] B. Ramsdell et al., “S/mime version 3 message specification,” RFC 2633, June, Tech. Rep., 1999.

[12] M. Abadi, M. Burrows, M. Manasse, and T. Wobber, “Moderately hard, memory-bound functions,” ACM Transactions on Internet Technology (TOIT), vol. 5, no. 2, pp. 299–327, 2005.

[13] X. Fan, M. Li, J. Ma, Y. Ren, H. Zhao, and Z. Su, “Behavior-based reputation management in p2p file-sharing networks,” Journal of Computer and System Sciences, vol. 78, no. 6, pp. 1737–1750, 2012.

[14] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputa-tion systems for online service provision,” Decision support systems, vol. 43, no. 2, pp. 618–644, 2007.

[15] Y. Wang and A. Nakao, “Poisonedwater: An improved approach for accurate reputation ranking in p2p networks,” Future Genera-tion Computer Systems, vol. 26, no. 8, pp. 1317–1326, 2010.

[16] H. Kun and W. Lu, “Research of trust model based on peer-to-peer network security,” in 2013 International Conference on Information Technology and Applications. IEEE, 2013, pp. 126–129.

[17] X.-Y. Ma, "The trust model based on punish mechanism in peer-to-peer network," in 2012 Fourth International Conference on Com-putational and Information Sciences. IEEE, 2012, pp. 1108–1110.
[18] S. A. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," arXiv preprint cs/0412017, 2004.
[19] K. Singh and H. Schulzrinne, "Peer-to-peer internet telephony using sip," in Proceedings of the international workshop on Network and operating systems support for digital audio and video. ACM, 2005, pp. 63–68.
[20] A. R. Butt, T. A. Johnson, Y. Zheng, and Y. C. Hu, "Kosha: A peer-to-peer enhancement for the network file system," Journal of Grid Computing, vol. 4, no. 3, pp. 323–341, 2006.
[21] N. S. Good and A. Krekelberg, "Usability and privacy: a study of kazaa p2p file-sharing," in Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 2003, pp. 137–144.
[22] G. Ramachandran and D. Hart, "A p2p intrusion detection system based on mobile agents," in Proceedings of the 42nd annual Southeast regional conference. ACM, 2004, pp. 185–190.
[23] A. Detsch, L. P. Gaspary, M. P. Barcellos, and R. N. Sanchez, "Flexible security configuration & deployment in peer-to-peer applications." in NOMS, 2006, pp. 209–219.
[24] J. Walkerdine and S. Lock, "Towards secure mobile p2p systems," in Internet and Web Applications and Services, 2007. ICIW'07. Second International Conference on. IEEE, 2007, pp. 6–6.
[25] C. Huitema, A. Gavrilescu, and X. Zhang, "Peer-to-peer name resolution protocol (pnrp) group security infrastructure and method," Jun. 27 2006, uS Patent 7,068,789.
[26] P. Dewan and P. Dasgupta, "Securing p2p networks using peer reputations: is there a silver bullet," in IEEE Consumer Communica-tions and Networking Conference (CCNC 2005). Las Vegas, Nevada, USA, 2005.
[27] B. Yu, M. P. Singh, and K. Sycara, "Developing trust in large-scale peer-to-peer systems," in Multi-Agent Security and Survivability, 2004 IEEE First Symposium on. IEEE, 2004, pp. 1–10.
[28] E. Palomar, J. M. Tapiador, J. C. Hernandez-Castro, and A. Rib-agorda, "Dealing with sporadic strangers, or the (un) suitability of trust for mobile p2p security," in Database and Expert Systems Applications, 2007. DEXA'07. 18th International Workshop on. IEEE, 2007, pp. 779–783.
[29] E. Damiani, S. D. C. Di Vimercati, S. Paraboschi, and P. Samarati, "Managing and sharing servants' reputations in p2p systems," IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 4, pp. 840–854, 2003.
[30] D. O. Rice, "A proposal for the security of peer-to-peer (p2p) networks: a pricing model inspired by the theory of complex networks," in Information Sciences and Systems, 2007. CISS'07. 41st Annual Conference on. IEEE, 2007, pp. 812–813.
[31] N. Saxena, G. Tsudik, and J. H. Yi, "Admission control in peer-to-peer: design and performance evaluation," in Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks. ACM, 2003, pp. 104–113.
[32] Y. Kim, D. Mazzocchi, and G. Tsudik, "Admission control in peer groups," in Network Computing and Applications, 2003. NCA 2003. Second IEEE International Symposium on. IEEE, 2003, pp. 131–139.
[33] K. Sethom, K. Masmoudi, and H. Afifi, "A secure p2p architecture for location management," in Proceedings of the 6th international conference on Mobile data management. ACM, 2005, pp. 22–26.
[34] K. Berket, A. Essiari, and A. Muratas, "Pki-based security for peer-to-peer information sharing," in Peer-to-Peer Computing, 2004. Proceedings. Proceedings. Fourth International Conference on. IEEE, 2004, pp. 45–52.
[35] Y. Takano, N. Isozaki, and Y. Shinoda, "Multipath key exchange on p2p networks," in Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on. IEEE, 2006, pp. 8–pp.
[36] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," ACM SIGOPS Operating Systems Review, vol. 36, no. SI, pp. 299– 314, 2002.