

# Peer to Peer Computing and Trusted Computing, an Odd Match or Made to Be?

Chaitanya Vinay Hazarey  
CSCI 530 Term Research Paper  
chaitanya.hazarey@usc.edu  
Computer Science Department, University of Southern California

*Abstract- A technology, created by some graduate students, back in 1970's, a technology rapidly gaining immense popularity, a technology heralded as the fastest growing internet application<sup>1</sup>. All these superlatives apply to the technology I am talking about called the Peer to Peer<sup>2</sup> computing or P2P file sharing. Although the technology is not new, but when applied to a certain situation, i.e. the sharing of files and information, the situation takes a rather interesting turn. The basic tenet of the P2P according to me can be summarized in the following quote by Tom Robbins, "Equality is not in regarding different things similarly, equality is in regarding different things differently.". The main strength of this technology is according to me is the empowerment of peers or individual machines. There has been immense research about this technology and it has basically given the world a proof of concept of its working. Now the thing remains is to formulate some business models around this technology. This paper plans to study the hindrances in the formation a viable business model around this technology and attempts to present a solution to the problems. A solution to some problems can some times come from the most unlikely of places. So it seems is especially true in the case of P2P techniques. I will try and focus on a particular group of open specifications<sup>3</sup>, their stated goal being the provision of a more secure computing platform, for the solution of problems found in the P2P systems.*

## I. INTRODUCTION

Peer to Peer<sup>4</sup> computing is often looked on as a stigma in the computing world. Many people cannot imagine a meaningful computational resource management scheme using P2P as a component. I plan to explore what exactly is the core concern(s) in this way of information sharing. And is there a way to remove the reservations against it and make it more secure and more reliable ? Is there some way that in which some recent technologies such as "Trusted Computing" services which can make P2P computing more secure and make it more mainstream ?

P2P computing is a unique model of computing, in which there is no distinction between a client and a server, or rather we can say that each node acts as a client to other servers and is itself a server (for other clients). The computing power and the bandwidth is shared and contributed by the contributing peers. Peers can be looked on as individual participants without any prior commitment or contract with the other participants in the network. They may be seen as volunteering

for a common or selfish interest. There are many moral and legal issues involved in harnessing the power of P2P paradigm. But keeping a very objective viewpoint we will not discuss them in this paper.

The P2P computing has been the favorite scapegoat of the Digital Rights Management and Copyrights Management concerned industries. But it posses some properties such as amazing resilience to failure and fast and effective regeneration of lost links, load balancing properties and a certain self healing properties. These factors and the success of some P2P computing systems such as SETI@Home [3] are forcing people to rethink their stances towards this technology.

Trusted Computing on the other hand is a promising field in the area of computer security and Digital Rights Management. It proposes security features built in both computer software and hardware. These two seem to be a very incompatible pair at first but when we explore the features Trusted Computing can lend to P2P computing, we see P2P computing as a very good content distribution mechanism (when augmented by the Trusted Computing Services). We will not focus on the moral or the legal aspects of the Trusted Computing, but continuing our neutral and objective approach throughout this paper we will try and focus our attention on the solving of P2P system's problems using this approach.

For the rest of the paper we will look into the P2P computing, see how it is structured and how it works. Then we will proceed towards the description and working of the Trusted Computing Platform and its Specifications. Finally we will take a look at the various issues in P2P computing and their respective solutions in TC.

## II. WHAT EXACTLY IS PEER TO PEER FILE SHARING ?

P2P started back in the 1970's by a couple of graduate students from the Duke University<sup>5</sup>. At first it was limited to sharing messages on bulletin boards, but none the less the technology did exist. In the 1999, Napster [1] [2] was the first application to recognize and implement the potential of this system of sharing information. The twist was that it no longer shared measly messages, but had evolved to share complete files. The main design idea behind the whole concept is that, in an event of data transfer, the only responsible parties are the Requestor (of data or file) and the Supplier (of data or

<sup>1</sup> Source: A Cnet news article, Title "Napster among fastest-growing Net technologies", by Rachel Konrad, [Online] Available: <http://news.com.com/2100-1023-246648.html>

<sup>2</sup> Called P2P hereafter.

<sup>3</sup> Source: TCG Specifications [Online] Available: <https://www.trustedcomputinggroup.org/specs/>

<sup>4</sup> Source: Peer-to-Peer from the Wikipedia, [Online] Available: <http://en.wikipedia.org/wiki/Peer-to-peer>.

<sup>5</sup> Source: Description and History of Usenet from the Wikipedia, [Online] Available: <http://en.wikipedia.org/wiki/Usenet>.

file). There exists a direct connection between the Requesting node and the Supplier node, as the two nodes are completely autonomous and independent, they can assume the roles of a Server (supplying the data or file), or a Requestor (receiving the data or file). Hence they are called Peers. But none the less, there is always a Supplier (Server) and a Requestor (Client) in the transaction. Here there is a complete absence of any central control or information repository. And there exists absolutely no middle men between to two nodes engaged in the transaction.

To understand the concept further and the properly study the existing systems we take the help of two classifications, one proposed by Lua et al. [4] and the other by Backx et al. [5]. Lua proposes the classifications based on the **organization of nodes in the network**, i.e. **Structured P2P** networks and **Unstructured P2P** networks. Structured P2P systems are those which contain a rigorous and a well defined structure to their organization. The distribution of information is also according to a scheme and information is maintained at a central location as to what information is where. The searches here in this type of a network are more structured and efficient in finding a data object even if only a single copy exists in the network. Such systems are typically based on the **Chord/Distributed Hash Table** mechanism [6]. Unstructured P2P systems typically will have a quite a few number of peers joining and leaving at any given moment of time. The organization is typically less structured and there is minimum central information maintained about the state of the data in the system. So for the location of data, in the system the query method used is typically flooding and random walks of the nodes to find information. This type of system is much suited in a very dynamic environment and also when the bulk of the queries are directed towards well replicated data.

Backx et al. further elaborates this according to the **distribution of control in the system and the organization of knowledge repositories across the network**. There are according to his classification three types of P2P architectures. First one being the **Pure P2P architecture**, this is the genre to which the old **Gnutella** [7] and **Freenet** [8] belong to. Here there is a total lack of central control or knowledge about the nodes. The data is typically stored and retrieved in the form of complete files. This design has the potential to provide total anonymity and resilience to failures but at the cost of performance and inefficient searches. The second type of architecture is the **Mediated architecture**, this type of system is characterized by the presence of a central knowledge repository which maintains the data locations and the particulars of peers across the network. Here we achieve efficient searches and also some performance gain at the cost of anonymity and making the knowledge repository a single point of failure. The most (in) famous of systems, Napster belongs to this category. The third type of architecture attempts to augment the model of P2P systems by drawing on the strengths of both systems. This is called the **Hybrid architecture**. This architecture is characterized by the formation of overlay network of peers called the “**ultrapeers**” or “**superpeers**” [9], which in turn control a small subset of “normal” peers. The communication

between the superpeers is pure P2P. So this architecture builds up on the search and data distribution efficiency of the Mediated architecture and the resilience and the dynamicity of the Pure architecture. The new Gnutella is an example of such an architecture. It attempts to choose the more efficient and better networked peers to be the ultrapeers. This gives the recognition and exploits the heterogeneity of the peers. This is an excellent scheme to organize peers in and is fast gaining popularity and acceptance due to the benefits like self healing (from Pure architecture), efficiency (from the Mediated architecture), robustness and autonomy.

So from this study of the P2P file sharing structures we can infer that this mode of information sharing contains many desirable features and more. The main areas in which the current P2P systems lag behind are the areas of Performance and Security. There has been quite a lot of research for the performance enhancement of the system and various new mechanisms have been proposed, tested and implemented. But there seems to be a sort of tradeoff between the two factors when considered in the context of the P2P systems. What this means is that performance is sometimes achieved at the expense of security and vice versa. As it is possible more stringent checks on contents and peers will result in lower peer contribution rate and slower searches and consequently slower downloads. The security aspect is the most important one to be considered if we propose to build a viable business model around it. We will address the various attacks that can be made on the P2P networks and the various security properties it violates.

But we first we need to take a look at the specifications and mechanisms which has been proposed by the Trusted Computing Group, and aimed at making the PC computing more secure. We will take a brief look at the architecture and the methods proposed, and will then evaluate the feasibility of applying this approach for the solution of various problems faced by the P2P systems.

### III. WHAT IS TRUSTED COMPUTING PLATFORM ALL ABOUT ?

Trusted Computing is a technology proposed by the Trusted Computing Group<sup>6</sup>, which is aimed at providing a more secure environment for computing. The Trusted Computing Group<sup>7</sup>, has drafted a set of specifications called the Trusted Computing Specifications<sup>8</sup>. These specifications are a set of rules which are enumerated by the TCG, and they apply to the hardware of a computing platform and also provide a set of rules which specify the nature of support the software (Operating System and all the software to be run on the machine) has to provide to the hardware. Here it is important to clarify the meaning of “Trust” in this context. is The word “Trust” here does not imply the meaning of trustworthiness in this context. It means that, given all the specifications are properly followed (and the proofs thereof

<sup>6</sup> Source: TCG FAQ [Online] Available: <https://www.trustedcomputinggroup.org/faq/>

<sup>7</sup> Hereafter referred to as TCG.

<sup>8</sup> Source: TCG Specifications [Online] Available: <https://www.trustedcomputinggroup.org/specs/>

provided to the appropriate parties), (this action) serves as a proof that the “whole” system can be trusted (to be free from interference, passive or active) by the writers (of the software you want to run) and the designers (of the systems). Trusted Computing<sup>9</sup> implements this technology of proof calculation with the help of hardware of the machine and the appropriate software.

TC attempts to form a chain of trust based upon the hardware components and extending to the software that you run on that PC. This technology takes full advantages of the recent advances in the field of Public Key Cryptography, Conventional Cryptography, Random Number Generation, Calculation of Cryptographic Hashes and Technological advances in the hardware. As the TC relies on the hardware to some extent for **the bootstrapping of trust in the PC**, it proposes the design of a Trusted Platform Module. This is the core “trusted” component of the whole system which resides on the hardware (near the Microprocessor of the machine). It is placed at a vantage point to as to have the ability to intercept the traffic perform measurements of variables at the lowest possible level. This silicon chip contains, Random Number Generators, key generator, key verifier, logger, data encryption and decryption engines, secure locations to store the data. It is obviously indelible and will record all the attempts to access it and circumvent the mechanisms in place.

This is the foundation of the “Hardware-rooted Trust<sup>10</sup>” of the whole mechanism. The chain of trust as stated earlier starts from the manufacturer of hardware. The manufacturer first certifies the hardware and the BIOS to be tamper resistant [10] and having the necessary security keys in place. This hardware and BIOS then in turn certify the code of the operating system that runs on it to be trusted, the operating system then vouches for the drivers and the various other system software running on it, in the end of the chain the operating system proves that the end user application software running is trusted and secure. This is a sort of layered mechanism which builds on the trusted-ness of the previous layers. Any violation of trust in any layer will render the whole system as untrustworthy. So if a software provider wants a proof of trust of the system, this whole chain will be followed the proof of trust will be presented to the provider of the software content. The other thing that this trusted-ness extends to the inter hardware communication channels as well to ensure that the information is not leaked in transit. This secure channel extends all the way to the end device which is in direct contact with the user (ex monitors, speakers etc.).

Keeping the above discussion in mind we can safely infer that Trusted Computing seems to be a very good and secure way to do computing. We will now explore the various security related problems in the P2P systems. And will try and argue if the Trusted Computing Platform can extend the model of P2P systems and augment the security side of it to make it a viable content distribution model.

#### IV. PROBLEMS IN THE PEER TO PEER FILE SHARING MODEL AND THEIR SOLUTIONS BY TRUSTED COMPUTING.

There are quite a few papers which express concern over the viability of the Peer to Peer systems as an effective content distribution mechanism [11] [12] [13]. And discuss the attacks and the security threats the P2P systems faces and (also) is the cause of. I will try to relate these concerns to the basic security principles and while stating the ways in which these concerns can be addressed by the Trusted Computing.

Following are the concerns which plague the P2P system and by the removal or the addressing of which the P2P system could be marketed as a secure system.

**Identity Theft, causes, effects and the solution:** Balfe et al. [14], points out that the main security concerns of this system are the Accountability and Anonymity. They propose that due to the fact that there is no control over the user’s identity and the peer has the freedom to assume various identities, thereby circumvent various security measures built in the system. This fact gives the peer an undue advantage in the form of assuming various false identities and harming the system by poisoning the data contents and providing false data. A particularly malicious peer may also assume the identity of other and wreak havoc in the system. As there is a lack of accountability, and there is no maintenance of an audit trail which is associated to a particular peer. The blacklisting mechanism is also not that effective. As there is always room for the peer to assume another identity and continue his malicious activities.

They further propose that this problem can be solved by using the already implemented Direct Anonymous Attestation in TC. So every time a peer claims that it is associated with a particular identity, it has to provide proof of the claim by supplying the appropriate DAA credentials. This basically binds the identity to a single peer, prevents a peer to assume multiple identities and also prevents identity theft.

Schechter et al. gives detailed explanation about the attacks on the P2P systems [10], their causes and the solutions using TC. They propose that TC will go a long way to make the P2P systems popular. The attacks have been explained regarding three security assets the attackers target. They are **Confidentiality, Integrity and Availability**. I will briefly restate them here.

They elaborate various attacks on the different security assets stated above. Confidentiality is under attack basically because of lack of mechanism to ensure that the communication channels are secure. The knowledge gained from these attacks is also used to arm the attacker with enough material to launch an attack on the Integrity and the Availability of the system. Some common ways to secure the channels such as cryptography are not much useful because of the existence of data pattern analysis. This evidence is enough to present as a proof in legal action and attract severe penalties for both the parties.

The main advantage of a P2P system is that all the membership is voluntary and the content is voluntary too. This being an advantage in some sense is a problem when the security of the network is concerned. The responsibility of maintaining the data and the programs is completely on the peer’s owner, and there is a lack of central verifying

<sup>9</sup> Hereafter referred to as TC.

<sup>10</sup> Presentation “Infrastructure for Trusted Computing”, in briefing to ACSAC, Dec 7, 2004. [Online] Available: <http://www.acsac.org/2004/workshop/Thomas-Hardjono.pdf>

authority. This gives the malicious user enough room to poison the network with inferior quality content or virulent content. There are systems in place which will prevent such users from accessing the system again, but they are based on very weak schemes. For example some are based on the handle (a pseudo name the user assumes in a P2P network) while some rely on the IP addresses. These are not enough to maintain a solid reputation mechanism, and to keep the malicious users out once they are proven to be so.

As we have stated earlier, the attacker might gather enough amount of knowledge about the system and the constituent software that eventually he will be armed with enough arsenal to attack the network as a whole. This may be achieved by writing customized malware which will ape the real program and will disrupt the real traffic of the network. Such "bot-nets", can be easily installed on unsuspecting user's machines. It is not hard to obtain the code for some of the P2P clients present, for the rest of them, it would be enough to do a statistical analysis of the data sent and received, and reverse engineer a solution (or the way in which the software works<sup>11</sup>). While this may be a great tool to learn about the system, it may be abused for more nefarious activities. There is a possibility of compromise of some super-peers (which are responsible for a subset of peers), to make them direct all the queries to a particular host for resolution [12], this may lead to overwhelming of a peer and subsequently lead to its shutdown. The magnitude of these P2P systems is quite large, at any given moment; there are thousands of peers and millions of queries generated. This fact makes these systems quite vulnerable to the attacks on their availability.

All the above attacks are quite possible and quite present in the real world P2P systems. The lack of assurance that these attacks can't take place is a major hindrance in bringing the P2P systems to the real world and making them economically viable solutions. So if we introduce the various services provided to us by TC, we may be able to address all these problems at a much more fundamental level. A technique called remote attestation [15], provided by the specifications of TC, will be especially of help here. It provides for the remote authority for a way to ensure that the machine they are trying to connect to, confirms to certain security standards. It provides a way to give proof about the authenticity of the hardware and the operating system (and other related software). This ensures that there is no virulent software or malware running on the machine and all its communications channels are tamper proof. These types of mechanisms give us tools to prepare a sort of reputation mechanism based on the previous activities of the peer. This will allow us to remove the malicious peer from the system or at least circumvent the area of its influence. This approach will address the availability and the integrity issues.

---

<sup>11</sup> OpenNap: Open Source Napster Server, Napster protocol specification, [Online] Available: <http://opennap.sourceforge.net/napster.txt>

## V. CONCLUSIONS

In this paper we have first explored the details about the P2P way of computing and information sharing. It contains many desirable properties and more. P2P system when evolved enough so as to combat the various security issues will have enough potential to base a business model on. Also it can serve as an efficient content distribution mechanism. The main issues we have dealt with and have studied in this paper are the Security issues in P2P systems and their resolution. The technology proposed by the Trusted Computing Group, TC promises to be a better and a secure way to do computation. It provides specifications for the "trust" in a system's hardware and software. The services provided by the TC are the exact ones which are needed by the P2P systems to establish themselves in the market as a viable technology. So from this discussion and reviews presented in this paper we can "trust" that P2P systems and Trusted Computing is a match made to be.

## ACKNOWLEDGMENT

This paper draws heavily on the discussions, lectures and presentations of Prof. Clifford Neuman presented in his class CSCI 530, Security Systems. It may inadvertently contain some materials provided in the classes [16]. I wish to thank him for his excellent support and guidance he has provided to me during the course of this class and during our discussions.

## REFERENCES

- [1] OpenNap: Open Source Napster Server, Napster protocol specification Current documentation for known client-server protocol, [Online] (last updated on March 12, 2001), Available: <http://opennap.sourceforge.net/napster.txt>.
- [2] Napster, From Wikipedia, the free encyclopedia, [Online], Available: <http://en.wikipedia.org/wiki/Napster>
- [3] Anderson DP, Cobb J, Korpela E, Lebofsky M, Werthimer D, "SETI@home: an experiment in public-resource computing", *Commun. ACM*, Vol. 45, No. 11. (November 2002), pp. 56-61.
- [4] Lua K, Crowcroft J, Pias M, Sharma R, Lim S, "A survey and comparison of peer-to-peer overlay network schemes", *Communications Surveys & Tutorials, IEEE (2005)*, pp. 72-93.
- [5] Peter Backx, Tim Wauters, Bart Dhoedt, Piet Demeester, "A comparison of peer-to-peer architectures", *Eurescom Summit 2002, Heidelberg, Germany*.
- [6] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications", *IEEE/ACM Transactions on Networking (TON) archive*, Volume 11, Issue 1 February 2003 pp. 17 - 32.
- [7] Gnutella Development Team, Gnutella development forum, the gnutella v0.6 protocol, [Online], Available: [http://groups.yahoo.com/group/the\\_gdf/files/\(2001\)](http://groups.yahoo.com/group/the_gdf/files/(2001)).
- [8] The Free Network Project, "FreeNet White Paper", [Online] Available: <http://freenetproject.org/papers/freenet.pdf>.
- [9] Beverly Yang, Hector Garcia-Molina, "Designing a Super-Peer Network", 19th International Conference on Data Engineering (ICDE'03) pp. 49.
- [10] Stuart E. Schechter, Rachel A. Greenstadt, Michael D. Smith, "Trusted Computing, Peer-To-Peer Distribution, and the Economics of Pirated Entertainment", The Second Annual Workshop on Economics and Information Security, College Park, Maryland, May 29-30, 2003.
- [11] Samsudin, A.T., Arshad, A., Abu Bakar, F., Mukthar, Z., "Promising or deceptive peer-to-peer computing?", *ELMAR*, 2005. 47th International Symposium, Date: 8-10 June 2005, pp: 175- 178.
- [12] Neil Daswani, Hector Garcia-Molina, Beverly Yang, "Open Problems in Data-Sharing Peer-to-Peer Systems", *Lecture Notes In Computer Science*; Vol. 2572 archive, Proceedings of the 9th International Conference on Database Theory, pp: 1 - 15, 2003.
- [13] Jung-Tae Kim, Hae-Kyeong, Park Eui-Hyun Paik, "Security issues in peer-to-peer systems", *Advanced Communication Technology*, 2005,

- ICACT 2005. The 7th International Conference on, Date: 21-23 Feb. 2005, Volume: 2, pp: 1059- 1063
- [14] Balfe, S., Lakhani, A.D., Paterson, K.G., "Trusted computing: providing security for peer-to-peer networks", Peer-to-Peer Computing, 2005. P2P 2005. Fifth IEEE International Conference on, Date: 31 Aug.-2 Sept. 2005, pp: 117- 124.
- [15] Ville Likitalo, "Remote Attestation and Peer-to-Peer Networks", Helsinki University of Technology, Laboratory of Information Processing Science,T-110.551 Seminar on Internetworking 2005.
- [16] Prof. Clifford Neuman, Class notes for CSCI 530, Security Systems, [Online] Available: <http://ccss.usc.edu/530/fall06/>.
- [17] Ian Foster, Adriana Iamnitchi, "On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing.", Peer-to-Peer Systems II, Second International Workshop, IPTPS 2003, Berkeley, CA, USA.