IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

nCORETech 2017

LBS College of Engineering, Kasaragod

Vol. 6, Special Issue 3, March 2017

Peer-to-Peer Networks, SORT model and Trusted Platform Computing

Prathyusha Purushothaman[1], Dr. Vinodu George[2]

PG Scholar, Computer Science Department, LBS College of Engineering, Kasaragod, India [1]

PhD, Computer Science Department, LBS College of Engineering, Kasaragod, India[2]

Abstract: We are today in the era in which communication and information sharing has become integral part. Thus more and more technological advancements are coming up. A peer-to-peer network is among them. While sharing any information to another person we first have this thought if our data will be safe with the person or not. Thus, came the concept of securing the data and how significant trust can be. Many trust models have been proposed timely. One among these models is the Self Organizing Trust (SORT) Model. Even though many attempts are still going on to improvise the calculations, SORT has been able to solve more issues than its contemporaries. We suggest this paper to improvise the existing SORT model accompanied with a hardware concept called the Trusted Platform Module (TPM), which was introduced by Trusted Computing Group. Since most of the vulnerabilities lie in the software part, it is believed that TPM can withstand such attacks.

Keywords: SORT, TPM, trustworthiness, trusted computing.

I. INTRODUCTION

Peer-to-Peer (P2P) network deals with communication between two computing systems. In general, P2P can be divided broadly into - centralized P2P and decentralized P2P. Here, we are dealing with decentralized P2P networks. This gives equal privilege to every computing device. In the context of P2P network, we name each computing device as peers. Since, these peers are meant to share information or specifically files and documents among them, one of the most important issues that come up is- Trust.

For the communication between systems, a definitive trust has to be established. If such a trust is established, then in long term, systems can share data with ease. Several trust and reputation models and issues are being analyzed. Some such as trust bootstrapping, trust evidence, trust assessment, second order issues, interaction outcome evaluation, punishment, reputation propagation, redemption, context awareness and rewarding, dynamic nature and trust type value and so on. Trust bootstrapping deals with the initial trust value assignment which is the value a trust builder assigns to trustee. "First impression is the best impression" and a wrong judgment results in bad transaction. Trust evidence can involve direct or indirect interaction between the one who trusts and the trustee. Second order issues are security threats that are prevailing in the P2P as well as other network environments, namely individual malicious person attack or group of persons with bad intention, collusion attack, Sybil attack or Impersonation, camouflage attack or on/off attack, trusted peer changing nature etc. There have been several solutions for each of these attacks. In some of the existing trust models wide coverage of all attacks not being carried out. Trust interaction evaluation is done by watch dog, centralized or decentralized node, Public key infrastructure (PKI), monitoring node, etc. The trust evaluation may be performed locally or globally. In certain models, good service or transaction is rewarded by providing weightage to the satisfaction factor or if trust level crosses threshold value. Diminishing effect deals with trust decay over a period. In trust models it has become necessary to include context awareness. Trust evolves over a period of time, hence the trust model should be a dynamic one. The trust value can be discrete or continuous, but continuous trust value is preferable over the discrete. Since a random number or value cannot be assigned to Trust, there must be some relevant metrics to measure the trust among systems. Metrics should have precision so that peers can be ranked according to trustworthiness. Interactions and feedbacks of peers provide information to measure trust among peers. Interactions with a peer provide certain information about the peer but feedbacks might contain deceptive information. This makes assessment of trustworthiness a challenge. In the presence of an authority, a central server is a preferred way to store and manage trust information. Since there is no central server in most P2P systems, peers organize themselves to store and manage trust information about each other.

Management of trust information is dependent to the structure of P2P network. In distributed hash table (DHT) - based approaches, each peer becomes a trust holder by storing feedbacksabout other peers [4][5]. Global trust information

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
nCORETech 2017
LBS College of Engineering, Kasaragod
Vol. 6, Special Issue 3, March 2017

stored by trust holders can be accessed through DHT efficiently. In unstructured networks, each peer stores trust information about peers in its neighbourhood or peers interacted in the past [3], [6], [7]. A peer sends trust queries to learn trust information of other peers. A trust query is either flooded to the network or sent to neighbourhood of the query initiator. Generally, calculated Trust information is not global and does not reflect opinions of all peers. Self-Organizing Trust model aims to decrease the malicious activity in P2P system by establishing trust relations among peers in proximity. Peers do not try to collect information from all the peers. Each peer develops its own local view of trust about the peers interacted in the past. In this way good peers form dynamic trust groups in their proximity and can isolate malicious peers.
In SORT, peers are assumed to be strangers at beginning. A Self-Organizing Trust model (SORT) that enables peers to create and manage trust relationships without using a priori information. Since pre-existence of trust among peers does not distinguish a newcomer and a trustworthy one, SORT assumes that all peers are strangers to each other at the beginning. Peers must contribute others in order to build trust relationships. Malicious behaviour quickly destroys such a relationship. A peer becomes an acquaintance of another peer after providing services i.e., uploading a file. If a peer has no acquaintance, it chooses to trust strangers. An acquaintance is always preferred over a stranger if they are equally trustworthy. Using a service of a peer is an interaction, which is evaluated based on weight (importance) and recentness of the interaction, and satisfaction of the requester. An acquaintance's feedback about a peer, recommendation, is evaluated based on recommender's trustworthiness. It contains the recommender's own experience about the peer, information collected from the recommender's acquaintances, and the recommender's level of confidence in the recommendation. If the level of confidence is low, the recommendation has a low value in evaluation and affects less the trustworthiness of the recommender.

A self-organizing trust model for P2P networks is presented in which a peer can develop trust relations without using a priori information. Trust metrics defined on service and recommendation trust contexts help a peer to reason more precisely about capabilities of other peers in providing services and giving recommendations. If all peers are behave good, reputation of a peer is proportional to its capabilities such as network bandwidth, average online period and number of shared files. In a malicious network, service and recommendation-based attacks affect the reputation of a peer. The service-based attacks are mitigated faster since victims quickly identify them. They gain a highest recommendation trust average and cause the victims to have the lowest average. Thus, they can continue to give misleading recommendations which can be stopped if a trusted third party is used. Defining a context of trust and its related metrics increases a peer's ability to identify and mitigate attacks in the context related tasks. Therefore, various contexts of trust can be defined to enhance security of P2P systems on specific tasks.

A peer may be a good service provider but a bad recommender or vice versa. A peer is assumed as trustworthy unless there are complaints about it. P-Grid provides decentralized and efficient access to trust information. Eigen trust uses transitivity of trust which allows a peer to calculate global trust values of other peers. An important aspect of trust calculation is to be adaptive for application dependent factors. Thus, SORT considers providing services and giving recommendations as different tasks and defines two contexts of trust: service and recommendation contexts. Information about past interactions and recommendations are stored in separate histories to assess competence and integrity of acquaintances in these contexts [1].

SORT defines three trust metrics. Reputation metric is calculated based on recommendations. It is important when deciding about strangers and new acquaintances. Reputation loses its importance as experience with an acquaintance increases. Service trust and recommendation trust are primary metrics to measure trustworthiness in the service and recommendation contexts, respectively. The service trust metric is used when selecting service providers. The recommendation trust metric is important when requesting recommendations. When calculating the reputation metric, recommendations are evaluated based on the recommendation trust metric. Hence, many trust establishing models have been proposed to guarantee the genuineness of peer nodes. Among many available Trust Models one that we are considering here is the Self ORganizing Trust (SORT) Model. It tends to solve many of the trust issues.

Issues identified within SORT model:

- Bootstrapping Issue
- No categorization of peers into similar peers or role based peers, local, global, community trusted peers are missing.
- In indirect trust experiences, referrals are not taken into account,
- Weightage for confidence as recommender is not used.
- Interaction outcomes when evaluated by node itself, misjudgement might occur.
- No solution for decay trust
- Changing of benevolent peers is not recognised and hence dynamic and context awareness factors are missing.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
nCORETech 2017
LBS College of Engineering, Kasaragod
Vol. 6, Special Issue 3, March 2017

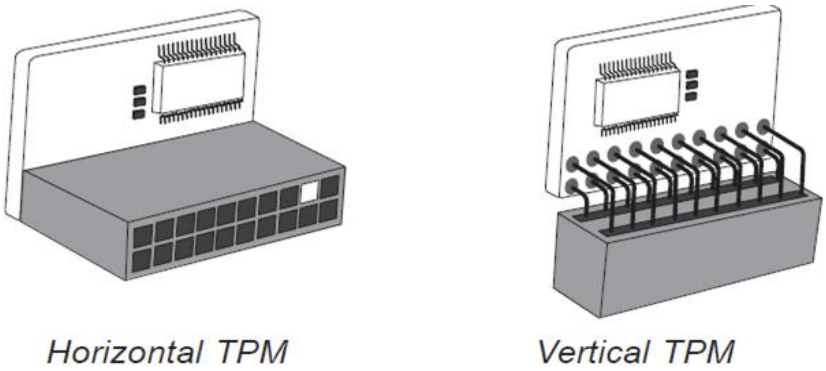

Fig.1.The TPM hardware chips

II. WHY TPM

With the use of a hardware technology like Trusted Platform Module some of these issues can be handled.
The TPM being hardware has very useful encryption property that is certified by a trusted certificate Authority (CA). It uses secret key, called Endorsement key, that is burnt into it during manufacturing.
Now there are a few reasons why we are opting for TPM.

- It is compatible to any of the Operating System.
- It works best in conjunction with other security technologies like Firewall, Antivirus software, Smart Cards, Biometric verifications.
- A Storage Root Key (SRK) is generated when user or administrator takes ownership.
- Its working does not affect the machine speed while securing the system.

One of the features provided by TPMs is certified migration. This process allows for one TPM to migrate an internal key to another TPM in such a way that this key is shared between the TPMs and yet is never available in a usable form outside the TPMs. The primary use for this kind of key is for one system to be able to transfer keys and a protected storage hierarchy from one system to another.

A virtual monotonic counter is a trusted counter than can be incremented but not reset back to any previous value. This security property is enforced by the TPM alone and does not require a trusted OS for this purpose. Among other interesting applications, virtual monotonic counters allow us to realize count-limited objects or "clobs" which are tied to a particular virtual monotonic counter. These can be n-time use encryption or signature keys, for example. The use of this key is tied to the counter which enforces that the key is not used more than n times.

A. Basic Components of TPM

A trusted platform must have special locations where sensitive data can be operated on.[9] It should not be possible for user programs to access these locations. These are called shielded locations and can be accessed only with a certain set of commands called protected capabilities. The
TPM implements protected capabilities for protecting shielded locations and for reporting integrity measurements. This can include additional functionality like random number generation, generation of cryptographic keys and sealing data to system state.

• Secure I/O: This component manages information flow across the communication bus. It performs encoding and decoding of information passed over internal and external buses. It routes messages to the appropriate TPM component.
• Key generation: This function is a protected capability and manages the generation of keys and nonce.
• Cryptographic engine: This component includes the SHA-1 engine, the RSA engine and the HMAC engine.
• Opt-in: This component allows the TPM to be disabled if necessary. The TPM isdisabled by default and the owner must enable it via the Opt-in.
• Random number generator (RNG): This is the source of entropy in the TPM.
• Platform configuration registers (PCRs): These are 20-byte values that are SHA-1 digests. The integrity metrics, obtained from measuring state, are stored in PCRs. Version 1.2 of the specification requires 24 PCRs.
• Monotonic counter: This hardware counter provides an ever-increasing value. Its value, once incremented, cannot be reverted to a previous value. The TPM must support at least 4 concurrent counters, although only one is active in any boot cycle.
• Non-Volatile storage: This holds persistent state information and identity information.
• Execution engine: This component runs the program code to execute TPM commands and ensures that operations are segregated and shielded locations are protected.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
nCORETech 2017
LBS College of Engineering, Kasaragod
Vol. 6, Special Issue 3, March 2017

There have been different versions of the TPM. The most significantly recognized ones are 1.1b, 1.2 and 2.0. Following are some significant features of these versions.

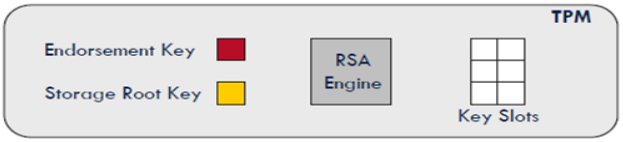

Fig.2.Initial Unowned TPM chip

It can be seen that the key slots are empty. These slots get filled when the user uses applications and Trusted Platform Module provides specific keys for each application.

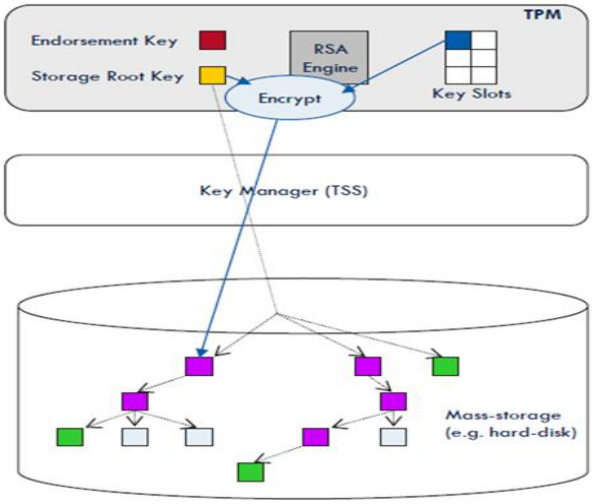

Fig.3.TPM allotting key slots to a specific application along with its encrypting values

III.BACKGROUND OF TRUSTED PLATFORM

The idea of using secure hardware to enhance security in computers has been around for a long time and definitions of what constitutes secure hardware abound. These devices can range from simple devices with limited computational power, like cryptographic smart-cards, to sophisticated high-end devices like secure co-processors that come with powerful processors and provide a high-degree of tamper resistance The trusted platform modules proposed by the TCG fall somewhere in between these two categories[8],[10]. These are tamper resistant devices with limited computational power that have a processor (not programmable by the end-user), a limited amount of non-volatile storage, hardware accelerators for performing cryptographic operations, and secure I/O interfaces. In a typical personal computer or larger server, a trusted platform module is a secure microcontroller that is securely bound to the motherboard and connected to the system using a low pin count (LPC) bus. The ability of a modern computing platform (like a PC or a laptop) to protect sensitive information through software alone is limited at best. Software cannot vouch for its integrity and even if software modifications can be detected locally, there is no mechanism for a remote user to be assured that the software is trustworthy. Trusted platform modules were developed with this simple threat model in mind and enhance security in modern computing platforms via the use of trusted hardware. Trust, in general, is defined as the expectation that a system will operate as it is expected to operate. In the context of trusted platforms, this is a guarantee by a known, trusted authority (usually the manufacturer of the platform) that a platform with a particular identity can be trusted to operate as expected. Trusted computing technology allows for execution of arbitrary software on the platform, while protecting information on the platform. Although executing programs have access to information protected by the trusted module, they cannot interpret the data since the data protected by the trusted module is encrypted.

The basic goal of trusted computing, as defined by the TCG, is to ensure trust in software processes executing on a computer, with the trust being rooted in hardware. Fundamentally, there is no way to distinguish arbitrary software from trusted software, other than to measure the executing software. Typically, this involves creating a hash of the executable and associating a secret with the measured hash. Trusted platforms provide a way to measure and store the resulting digest in a tamper-resistant environment. When queried by a remote party, a trusted platform has a way to provide proof of its capabilities as well as provide the measured digests to the remote party in an assured way.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

nCORETech 2017

LBS College of Engineering, Kasaragod

Vol. 6, Special Issue 3, March 2017

The TPM architecture includes non-volatile storage for storing cryptographic keys, a random microcontroller-number generator, a cryptographic engine that can perform cryptographic operations like RSA and SHA-1 and an I/O component that manages communication. Over the I/O bus. The added capabilities proposed for systems incorporating a trusted platform module include protected storage, cryptographic processing, integrity measurement and reporting, sealed storage, and remote attestation. As specified by the TCG, a trusted platform must have certain components and provide certain functionality in compliance with their specifications.

It is important to note that the TCG specifications are not constant or "set in stone"; they are evolving based on feedback from users with new functionality being added. To date, there have been two main versions, with various revisions of specifications with minor changes in each revision. Version 1.2 of the TCG specifications improved upon the first version 1.1 (also called 1.1b in various sources). The main changes between the two versions include support for certifiable migrate-able keys (CMKs), changes to the delegation model, improvement to security in the way data is sent to the TPM, specification of non-volatile (NV) storage and changes to address privacy concerns. We highlight the relevant changes in the appropriate sections.

IV. WORKING

The Trusted Platform Module works on the policies defined by the service provider or the manufacturer of the chip. Hence, user has less responsibility towards many of its functioning. So in order to make it more user friendly, it can be enhanced to a level that a user-defined policy can be defined. Here, for convenience we define the policy according to which the Peer-to-Peer network in congruence with the Self ORganizing Trust Model generates a value that means to be the trust value each peer calculates for every other interacted peer. Next, the Trusted Platform Module evaluates or checks upon the trust value. This value will need to be compared with a threshold value that TPM maintains. If TPM finds it all right, it establishes the connection with that peer. Hence, this helps in increasing the trustworthiness to the next level.

V. CONCLUSION

The TPM being a hardware component it would not be affected by the usual software based attacks that occur during peer-to-peer communication. The TPM provides uniqueness and genuineness to peers. It is although still to be considered that the initial connection among peer devices has to be based on a risk to trust the stranger. The physical access to the device or administrator privilege might be the only risks that TPM becomes vulnerable.

Other than this issue, TPM guarantees to provide good and reliable security to peer-to-peer communication.

ACKNOWLEDGMENT

I am grateful to the faculty of Computer Science Engineering Department of our college, LBS College of Engineering, Kasaragod, for all the valuable suggestions and ideas given in publishing this paper. I am obliged to thank our principal sir, for supporting us in this noble path. I am thankful to all the people who cherished the ideas and encouraged me to go forth.

REFERENCES

[1] Ahmet Burak Can, Bharat Bhargava, "SORT: Self-Organizing Trust Model for Peer-to-Peer Systems" IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 1, pp. 14-27, January/February, 2013.

[2] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," Proc. 10th Int'l Conf. Information and Knowledge Management (CIKM), 2001.

[3] F. Cornelli, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servants in a P2P Network," Proc. 11th World Wide Web Conf. (WWW), 2002.

[4] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The (Eigentrust) Algorithm for Reputation Management in P2P Networks," Proc. 12th World Wide Web Conf. (WWW), 2003.

[5] L. Xiong and L. Liu, "Peertrust: Supporting Reputation-Based Trust for Peer-to-Peer Ecommerce Communities," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 7, pp. 843-857, July 2004.

[6] A.A. Selcuk, E. Uzun, and M.R. Pariente, "A Reputation-Based Trust Management System for P2P Networks," Proc. IEEE/ACM Fourth Int'l Symp. Cluster Computing and the Grid (CCGRID), 2004.

[7] R. Zhou, K. Hwang, and M. Cai, "Gossiptrust for Fast Reputation Aggregation in Peer-to-Peer Networks," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 9, pp. 1282-1295, Sept. 2008.

[8] searchnetworking.techtarget.com/peer-to-peer.

[9] Trusted Computing Group official website- https:// www.tcg.com/.

[10] Chris Burnett, Timothy J. Norman, Katia Sycara, "Bootstrapping Trust Evaluations through Stereotypes", Proceeding of 9th international Conf. on Autonomous Agents and Multi agent Systems, May 10-14, 2010,Toronto,Canada, pp. 241-248.