ISSN 2229-5518

Trust Enhanced Authorization for Distributed Systems

Priyanka Dadhich, Dr. Kamlesh Dutta, Dr. M.C.Govil

Abstract— The trust –management approach to distributed system security is developed as an answer to the inadequacy of traditional authorization mechanism. The subjective concept of trust not only enables users to better understand the paradigm of pervasive computing, but also opens new direction of research for solving existing problems such as security [8], management of online communities or e-services lifecycle .This paper specifies research issues in the areas of authorization and trust in distributed environments involving mobile networks, pervasive and ubiquitous computing networks . We here discuss the notion of trusted computing and examine existing authorization mechanisms and their inadequacies. Next we define a logic program based languages and policies that facilitate the modeling process.To the end various approaches to trust enhanced security for overall authorization security in distributed systems are discussed.

Index Terms— Distributed, trust management, trusted computing, trust enhanced security, subjective trust

1 INTRODUCTION

TRUST based security models have shown the potential to overcome the drawbacks of traditional security models by ensuring a higher level of trustworthiness of authorized entities and thus raising the security levels.

The paper lays emphasis on the design and do management of authorization policies for distributed applications and introduces the notion of trust enhanced authorization to improve security decision making. Section two confers design principles and architectural frameworks for distributed authorization. Section three examine existing authorization mechanisms with their inadequencies. Section four explains the trust –management approach as an answer to inadequacy of authorization mechanisms by exploring some trust management inference engines . Section five discusses authorization policies and languages for trust modeling. Section six concludes and section seven puts forward some future prospects to enhance authorization using trusted platforms in distributed applications.

Authorization in distributed system is called distributed authorization. It is much richer than authentication both in terms of types of privileges required, its nature and its degree of interaction between participating entities. In earlier times, considerableefforts have been spent on formalizing security protocols and access control schemes for general distributed systems that include authentication logic and access control calculus by Abadi et al [36,4], a logic language for authorization specifications proposed by Jajodia et al [10], an access control policy description language proposed by Kurlowski [8] and Levier et al [6]. But these models combining authorization and authentication did not approach trust directly but rather deal with trust in an indirect way for identifying security flaws in the existing security protocols.

- *F.A. Author is Research Scholar with the Department of Computer Science & Engineering ,National Institute of Technology, Hamirpur (India). E-mail: prynkmshr@ gmail.com.*
- *S.B. Author is Associate Professor with the Department of Computer Science & Engineering, National Institute of Technology, Hamirpur(India) E-mail: kd@nitham.ac.in*
- *T.C. Author is professor with the Department of Coputer Engineering, Malviya National Institute of Technology,Jaipur and presently working as the Principal Govt. Women Engineering College ,Ajmer (India) E-mail:govilmc@yahoo.com*

2 DESIGN PRINCIPLES FOR DISTRIBUTED AUTHORIZATION(DA)

1. Designing of DA can only be accomplished by designing appropriate authorization attributes.
2. Designing should involve authorize information in the security service. Here security mechanisms are required to support these security service and the authorities involved in the management of the service[13].
3. Designing involves the locations where authorization checks can be made . These are:
   a. CoarseLevel Check:These determine whether access to the application is allowed or not.
   b. Function Access check: It is made on the type of function or operation being requested.
4. Designing of Distributed Authorization Service basically involves design of two distinct stages:
   a. Administration Design Phase: Involves design of facilities and services for the specification of authorization policies[15], updating and deleting of policies and their administration.
   b. Runtime or Evaluation Phase:It is concerned with the design of the use of these authorization policies in the evaluation of the access requests .

IJSER © 2011

Authorization Architecture Frameworks

- Authorization Architecture (AA) should involve to locate the static and generic authorization information i.e. responsible for a collection of clients and server principals[11].
- Frameworks should involve the dynamic and specific authorization information to be located near the target enabling the target system authorities to be involved in their management.
- These specific and dynamic authorization information needs to be " pulled" at the time of the decision process.
- Authorization frameworks consists of components like administration component where the authorization policies[6] are entered and stored in one representation and a runtime evaluation component that stores the authorization rules at a different representation for runtime access[40].

3 Inadequacies with Security Mechanisms

One security mechanisms used in Operating System is the ACL(Access Control Lists). This ACL is a list describing which access rights a principal has on an object(resource). Foe eg: UNIX file system "permissions" mechanisms is essentially an ACL. But unfortunately these ACL 's are inadequate for distributed systems(DS). These are:

1. Authentication : In DS, identity of principal is not known but known in OS. Since authentication is accomplished by username/ password mechanisms so this simple password-based protocols are inadequate in network computing environments[4]. Here simple eavesdropping can destroy security.
2. Delegation: Delegation enables decentralization of administrative tasks. It is needful for scalability of DS. In DS, policy(or authorization)[15] are specified at the last step in the delegation chain(the entity enforcing policy) in form of an ACL. This leads to inconsistencies among locally specified sub-policies[45] .
3. Expressibility and Extensibility: ACL approach do not provide sufficient expressibility or extendibility[24] . Hence may security policy elements that are not directly expressible in ACL form should have to be hard-coded into applications. Hence whenever there is change in security policy it often requires reconfiguration, rebuilding and rewriting of applications.
4. Local Trust Policy: Since the number of administrative entities in a DS are very large so each entity is given a different (local) trust model to be used by different users and by other entities.

For example: System A may trust System B to authenticate its users correctly but system A do not trust system C but system B trust system C.

All above security mechanisms are insecure, inadequate and non-scalable authentication mechanisms that are used in conjunction with ACLs. All these unintuitiveness and problematic mechanisms are in use because of the lack of alternatives that suit to DS.

4 Trust Management

The term 'trust management' was first introduced by Blaze et al [5] { role of trust management in security} . It is a unified approach specifying and interpreting security policies, credentials and relationships that allows direct authorization of security critical actions. These trust-management approach developed as an answer to the inadequacy of previous authorization mechanisms.

Trust Management system combines the notion of specifying security policy with the mechanisms for specifying security credentials. Credentials describe specific delegation of trust among public keys that bind keys to names, to perform specific tasks. These system supports delegation, policy specification, refinement at the different layers of a policy hierarchy. So, the system solves the consistency and scalability problems present in ACLs. Role of various components in Trust Management Architecture are:

1. Trust Manager: key component of proposed architecture that provides trust management services.
2. Trust Inference Engine: built on subjective logic primitives[30] .
3. Trust Policy Base: contains established trust relationships.
4. Trust Update: dynamically update the trust relationships in the trust base.
5. Trust Decision: provide trust decision from an owner host to requesting entities by preparing an itinerant computation.Trust decisions come from a set of trust based on initial set of trust relationships, recommended trust from others and observations of trust related actions over time[10].

Recommendation Protocol:These protocols are initiated by trust manager in the event of seeking trust information from its trusted entities about other unknown hosts[15]. This protocol helps to maintains a list of hosts (in its trust base) that are trusted for making recommendations.Recommendation is simply an attempt at communicating a party's reputation from one community context to another[20]. A poor recommendation may be detriment-mental to one's reputation and there is no separate term for "negative recommendation".

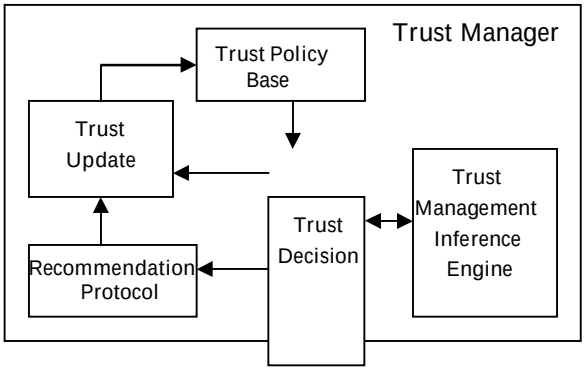

Fig.1 Trust Management Architecture

ISSN 2229-5518

4.1 Trust Management Inference Engine(TMIE)

It is a separate system component that takes input, outputs a decision about whether compliance with policy is proven or not and if not proven then outputs some additional information detailing how to proceed.

TMIE avoid the need to resolve "identities" in an authorization decision. These engines express privileges and restrictions in a programming language allowing for increased flexibility and expressibility and standardization of modern, scalable security mechanisms.

4.2 Designing of Trust Management Inference Engine

- Design should lay principals for defining proof of compliance.
- There should be some language or notations to express the policies and credentials.

4.3 Tools designed to avoid inadequacies in Distributed Authorization

1. PolicyMaker: It was the first tool for processing signed request embodying the trust management. It addresses the authorization problem directly rather than handling the problem indirectly by authentication or access control. Credentials and policies of PoylicyMaker are fully programmable and so called "assertions".

   PolicyMaker is a trust management application that specifies what a public key is authorized to do[22]. PolicyMaker system is essentially a querry engine tool that can either built into applications or run as a daemon service.
2. KeyNote: KeyNote[4] [10]. It has same design principals as Policymaker. Keynote uses credentials that directly authorize actions instead of dividing the authorization task into authentication and access control as in PolicyMaker.

   In KeyNote, standardization and ease of integration is developed to give applications. So, KeyNote assign more responsibility to Trust mananagement engine and less function to calling application. By fixing a specific and appropriate assertion language, KeyNote goes further than PolicyMaker toward facilitating efficiency, interoperability and widespread use of written credentials and policies.
3. REFEREE: This system fully supports programmability of assertions(i.e. policies and credentials) just like PolicyMaker. REFREE execution environment allows assertion programs to call each other as subroutines and to pass different arguments to different subroutines[18]. While PolicyMaker execution environment requires that each assertion program write anything that it wants to communicate , on a global blackboard i.e. seen by all other assertions. Refree system supports a more complicated form of inter-assertion communication than PolicyMaker.

4.4 Application Areas of Trust Management System

- In active networks
- In Mobile Code security
- In Access control Distributions

5 Authorization Policies and Policy Languages

A fundamental objective of any authorization system is to enable , to represent and to evaluate a range of access policies that are relevant and required. These policies capture the authorization requirements of the distributed applications.

Policy languages are useful in separating out the policy representation from policy enforcement. Some languages given in [8,10,11]{authorization and trust enhanced security for DA}are solely dedicated for specifying authorization policies. Languages discussed in [6,7] are mathematical logic based, some are graph based and some languages[8,9,10] are programming based. A standard policy language is useful for interoperability between different systems and applications.

Policy language's such as [18, 19, 20]{trust magmt survey} make it possible to automatically determine whether certain credentials are sufficient for performing certain actions or not to authorize the trustee. One trust management framework called Sultan trust management include a language for describing trust and recommendation relationships in the system. Constraints can easily be attached to these relationships and through them , the relationships can be connected to the Ponder policylanguage[22]{trust mangmt survey}.

Sufficient flexible policy system provide the backbone for a trust management system. Tonti et al[21] compare three languages for policy representation and reasoning[23]{. KAoS[24, 25] , Rei[21] and Ponder [16]are used as the basic languages for sketching some general properties desirable in future work on policy semantics.

5.1 Features of Policy languages

- Policy Language should deal with expression and do structuring of complex and dynamic relationships.
- Languages should be simple enough to enable the administrators and policy setters to use the language in specifying their policies.
- Language should have significant expressive and analytical power to represent and evaluate a range of policies used in practical systems.

5.2 Advantages of Policy Languages

- Use of these languages helps the administrators to save time and money because they are not required to rewrite their policies in many different programming languages.
- Developers are not require to invent new policy languges and write code to support them, so time is saved for developers.

- If policy languages are standardized , there are good opportunities for emerging good tools for writing and managing policies for a policy language.

5.3 Authorization Policies

These policies can range from simple identity based to complex dynamic and collaboration policies[24][12]. Some commonly used access policies are:

- Identity based policies
- Group based policies
- Role based policies
- Delegation policies
- Static separation of duty policies
- Dynamic separation of duty and Chinese wall policies
- Joint action policies
- Collaboration access policies

6 Approaches to Trust Enhanced Security

Trust enhanced security services require some form of "trusted " authorities to establish and manage "trust" between the mutually suspicious entities A and B over an untrusted network.

For authorization services, we have trust management components, authorization policies and mechanisms. Though the term trust is being used around many decades in different disciplines, but in security the concept of trust came in late 1970's.

With the development of TCSEC(Trusted Computer System Evaluation Criteria)[26] , trust is used in the system's model , design and implementation for its correctness and security. Afterwards, came TCB(Trusted Computer Base)that encapsulates all the security relevant components i.e. both hardware and software that are necessary for enforcing security policies in a system. Trust is the firm belief in the competence of an entity to act dependently, securely and reliably within a specified context[27].

6.1 Trust Notions

Trust provides better understanding of security and privacy problems.

- It acts as centralized control in a system.
- It issues resources to build reputation.
- It performs separation of concern.

Trust records feedback about the security evaluations of other nodes. Trust management enables the trust system to track the behaviour of each node and make corresponding reactions to the tracked behaviours. Trust management can establish a set of effective rules to make a reliable analysis of certain suspicious nodes.

6.2 Concept of Trust Management

Trust Management focuses on designing languages, compliance checkers, identifying applications and building practical toolkits. Beth et al [20] is one of the earliest trust models for authentication in distributed system focusing on relationship modeling whereas Abadi et al [11] {same} presented a modal logic based trust model for modeling distributed authentication and access control. Blaze et al [12] proposed a new well-known trust management system. The common shortcoming of these models is that they did not address the trust based on behavior or past experiences dynamically. Lin and Varadharajan [13] developed trust model for agent security management, but this model did not taken into account security risks that itself trust model has brought. So, all above factors, lead to the research for development of trust enhanced security models for distributed systems.

Later, Kagal et al [23] { a trust based security system for ubiquitous and pervasive computing} presented an architecture based on trust management applicable to distributed system and towards pervasive computing environments. This trust based architecture has a security policy i.e. responsible for assigning credentials to entities, delegating trust to third parties and reasoning about user's access rights. Ngai and Lyu provide a public key authentication service based on a trust model to monitor malicious and colluding nodes. This model allows mobile nodes in distributed system to monitor and rate each other with an authentication metric. The trust value can be updated in conjunction with public key certification. Zhu et al attempt to establish a secure route from a source node S to a designated node D, and provide an approach to calculate the trust value by applying a delegation graph. The mapping between a delegation edge and an authenticated transition graph is used to compute the trust value based on the transitive property.

6.3 Trust Management Authority

Below architecture is "rule-based" and "event-based" architecture.Here rules are used to define the policy of the trust management authority and categorize events that may occur in transactions. This architecture is adaptable to various domains of service oriented applications.

For provision of security services, trusted authorities such as authorization server and authentication server are involved that provide complete trust. For example: if entities A and B trust the authorization server (AS), this server will perform functions of A and B correctly and honestly. This AS will keep the authorization policies securely , perform authorization checks correctly and ensure that software of AS is free from any malicious software.Trust management system such as[31,32,33,34,35] are designed to support specification, acquisition, revocation, degradation and evolution of trust according to some model. It is the unified approach for specifying and inter interpreting security policies, credentials and relationships that allow direct authorization of security-critical actions[31].

Examples as described above: Some automated trust management systems are: PolicyMaker[23], KeyNote[9], REFEREE[17] being delegated.

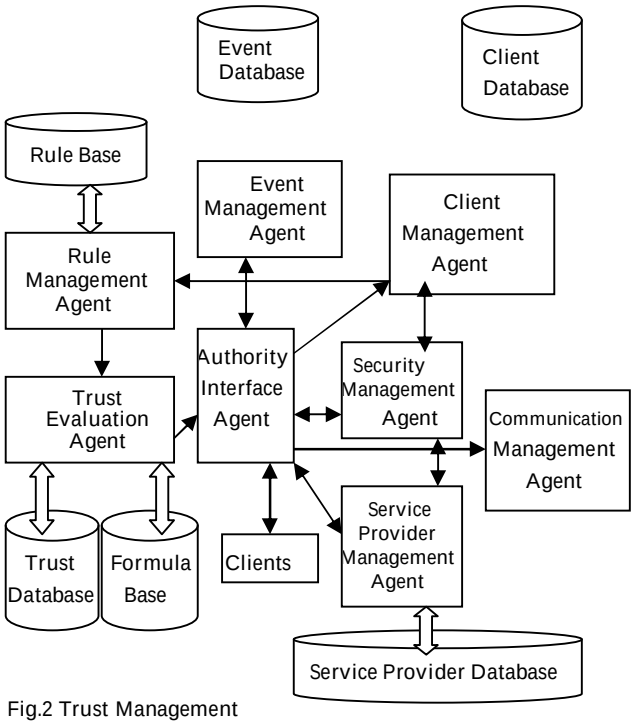

Fig.2 Trust Management Authority

6.3 Hybrid Trust

It is a composite trust relationship formed by combining hard and soft trust.

1. Hard Trust: Denotes the trust relationships that can be derived from the underlying cryptography based security mechanisms such as digital certificates and cryptographic checksums.

   These trust relationships indicate one agent host's belief in another in terms of authentication of the relevant host's identity (i.e. authentication trust)and the belief in the benevolence and competence of another host in producing good code (i.e. code trust) and the belief in the honesty and faithful and competent execution of the task requested by a visiting mobile agent i.e. migrating node or program called Execution Trust [3].

   Benefits of hard trust Models:

- Enable trust to be extracted from the security mechanism: by extracting trust from security mechanisms, we are able to find actual trust requirements of the underlying security mechanisms that helps us to make more effective security decisions.
- Enable categorization of hard trust related security mechanisms: using the hard trust notion , we can determine a range of hard trust mechanisms that can process and manage the hard trust information and helps to build trust models that can work with security models effectively.
- Enable trust management and its integration with the underlying security mechanisms for enhancements of security performance with help of hard trust models. Trust management systems can be designed that helps to feedback the trust decisions back to the underlying security mechanisms for performance enhancements.

2. Soft Trust:Soft trust is based on trust relationships derived from localized and external observations of system entity behavior[1]. These trusts are obtained through social control mechanisms such as direct observations, recommendations or combination of both.

   Many trust models [42,43] } are taken into soft trust models. Examples are subjective logic based trust model developed by Josang [30] and classical model of Beth et al [49].

   Benefits of Soft Trust Models:

- Social control principles are extensively studied in soft trust models so as to do research and to develop counter measures for malicious behavior in general distributed systems [19,6,18].
- It gives linking between behavior and evidence through mapping .
- By help of trust management operations, these soft trust models can calculate dynamic trust valuations based on the opinion calculus which is used for flexible trust decision making based on the specified thresholds for different trust requirements, in the form of several trust enhanced security protocols[34].
- Through these trust management protocols , the operations of recommendation based trust update and the end of transaction trust update make the distributed trust management possible.

7 Conclusion

In this literature , we have addressed some research issues in areas of authorization and trust in distributed environments. Some key design principles, policy language's and mechanisms, are discussed for the development of distributed authorization service. Trust management authority and hybrid trust concepts are explored to outline an idea for enhancing security concerns in distributed systems.

8 Future Work

With the development of term TCPA[39](Trusted Computing Platform Alliance){authorization and trust enhanced security for distributed application}, currently known as TCG (Trusted Computing Group) lead to the discovery of trusted platform technology comprising of a hardware based subsystem devoted to maintaining trust and security between machines. With the help of the availability of

trusted platform[33] and its characteristics any two entities that want to communicate with each other, has to go through trust determination phase before performing authorization at the beginning of the authorization process.

This above scheme can be extended to transfer authorization policies between two authorization server systems in two different domains. We currently need to develop such a distributed Authorization service on trusted platforms[39]. Also need to develop an application i.e. showing secure access of its operations using trust enhanced distributed authorization service[38]. Example of applications can be any military application, network management operations, healthcare applications or any e-commerce applications or any other[2].

REFERENCES

1. Heather, J., Hill, D.,:I'm Not Signing That! In Dimitrakos, T., MArtinelli, f., eds.: proceedings of the 1st Int'l Wksp on Formal Aspects in security and trust (FAST 2003), Pisa, Itlay(2003)71-81.
2. Ishaya, T., Mundy, D. P.:Trust development and management in virtual communities. In Trust Management : 2nd international conference ,itrust 2004, Oxford 2004.
3. Rindeback, C., Gustavsson, R.,: Why Trust is hard-Challenges in e-mediated services. In: Proceedings of the 7th Int'l wksp on Trust in Agent societies, New York, USA ,2004.
4. Lampson. B., Abadi, M., Burrows, M., Wobber, E.,: Authentication in distributed systems: theory and practice. ACM Trans. On Computer Science 10(4),1992.
5. Lampson. B., Abadi, M., Burrows, M., Plotkin, G.,: A calculus for access control in distributed systems. ACM Trans. On programming languages and systems 15(4),1993.
6. Jajodia, S., Samarati, P., Subrahmanian, V.S.: A logic language for expressing authorizations. In: Proc. IEEE Symp. On research in security and privacy, pp.31-42, 1997.
7. Maurer, U.,: Modelling a public key infrastructure. In : Martella, G., Kurth, H., Montolivo, E., Bertino, E.(eds) ESORICS 1996. LNCS, vol 1146 Springer , 1996.
8. Levien, R., Aiken, A.,: Attack –resistant trust metrics for public key certification. In: Proceedings of 7th USENIX security Symposium , 1998.
9. M. Blaze, J.Feigenbaum, J. Ioannidis and A. Keromytis. The KeyNote Trust-Management .Work in Progress, http://www.cis.upenn.edu/angelos/keynote.html.
10. M. Blaze, J.Feigenbaum, J.Lacy. Decentralized Trust Management. In Proc. of the 17th Symposium on security and Privacy, pages 164-173. IEEE Computer Society Press, Los Alamitos,1996.
11. V. Varadharajan, C. Crall and J. Pato,"Authorization for Enterprise wide Distributed Systems" Proceedings of the IEEE Computer security Applications Conference, ACSA'98,1998 USA.
12. M. Hitchens and V. Varadharajan, "ower: A language for Role Based Access Control" proceedings of Int'l Wksp on Policies for Distributed Systems and Networks,UK , 2001pp 88-106.
13. S. Indrakanti, V. Varadharajan , M. Hitchens and R. Kumar,"Secure Authorizations for Web Services" Proceedings of the 17th IFIP Conference on Data and Applications Security, USA, 2003.
14. S. Jajodia, P. Samarati and V.S. Subrahmanian," A Logical Language for Expressing Authorizations',Proceedings of the IEEE Symposium on Security and Privacy, USA, 1997.
15. Y. Bai and V. Varadharajan ,' A logic for State transformations in Authorization Policies' Proceedings of the IEEE Computer Security Foundations Wksp, USA, 1997.
16. N. Damianou, N. Dulay, E. Lupu and M. Sloman,"the ponder Policy specification Language', proceedings of Int'l Wksp on Policies for Distributed systems and networks, UK, 2001, pp 88-106.
17. Chu, Y. H.., Feigenbaum, J., LaMacchia, B., Resnick , P., Strauss, M.: REFEREE: Trust Management for web Applications. Computer Networks and ISDN systems 29 (1997) 953-964.
18. Blaze, M., Feigenbaum, J., Keromytis, A. D..: KeyNote: Trust management for public-key infrastructures(position-paper) In: security protocols:6th Int'l Wksp , Cambridge, UK, April 1998. Proceedings. Volume LNCS Springer- Verlag(1998) 59-63.
19. T. Grandison and M. Sloman. A survey of trust in internet application. IEEE Communications Surveys, 2000.
20. T. Grandison and M. Sloman. Specifying and analyzing trust for Internet applications. In : proceedings of 2nd IFIP Conference on e-commerce, e-business , e- government 13e2002, Lisbon, Portugal 2002.
21. Tonti, G., Bradshaw, J. M., Jeffers, R., Montanari, R., Suri, N., Uszok, A.: Semantic web languages for policy representation and reasoning: A comparison of KAoS, Rei and Ponder. In: The Semantic Web – ISWC 2003. Vol LNCS 2870 / 2003.419-437.
22. Damianou , N., Dulay, N., Lupu, E., Sloman, M.: The Ponder policy specification language. In: Wksp on Policies for Distributed System and Networks HP Labs Bristol 29-31 Jan 2001. Vol 1995, 2001.
23. Kagal, L., Finin, T., Joshi, A., : a policy language for a pervasive computing environment. In proceedings of tenth Knowledge Acquisition fpr knowledge-based system wksp,1995.

24. Uszok, A., Bradshaw, J. M., Jeffers, R. : KAoS: A Policy and domain services framework for grid computing and semantic web services. In: Trust Management : Second Int'l Conference, itrust 2004, Oxford, UK, March 29-April1, 2004. Proceedings. Volume LNCS 2995 / (2004) 16-26.
25. Bradshaw, J. M..: KAoS: An open agent architecture supporting reuse, interoperability and extensibility . In : Proceedings of 10th Knowledge Acquisition for Knowledge-Based Systems Workshop(1995).
26. Dept. of Defense, " trusted Computer System Evaluation Criteria",(TCSEC), DoD5200.28 STD Dec. 198.
27. L. Kagal, T. Finin, A. Joshi. Trust based security in pervasive computing environments, Computer 34(2001) 154-157.
28. H. Zhu, F. Bao, R. H. Deng. Computing of trust in wireless networks, In: proceedings of IEEE 60 th Vehicular technology Conference 2004, pp 2621-2624.
29. T. Grandison and M. Sloman. A survey of trust in internet application. IEEE Communications Surveys, , 2000.
30. A. Josang . A logic for uncertain probabilities . Int'l journal of uncertainty, Fuzziness and knowledge based systems 2001.
31. G. Zacharia and P. Maes. Trust management through reputation mechanisms. Applied Artificial Intelligence,2000.
32. C. Castelfranchi and R. Falcone. Principles of trust for mas: cognitive anatomy , social importance and quantification. In Demazeau, y. (ed) proceedings of the 3rd int'l Conference on Multi-Agent systems, IEEE Computer Society, 1998.
33. C. Lin Trust Enhanced Security for MA, PhD thesis, Macquarie University, August 2006.
34. C. Iin V. VAradharajan, Y. Wang and V. Pruthi. Trust enhanced security for MA. In 7th int'l IEEE conference on e-commerce technology 2005, IEEE Computer Society Press 2005.
35. B. Yu and M. Singh . A Social mechanisms of reputation management in electronic communities. In M. Klusch and L. Kerschberg, editors, CIA-2000 Wksp on Cooperative Information Agents, 1860 of LNAI, Springer, 2000.
36. B. Lampson, M. Abadi, M. Burrows and E. Wobber, Authentication in distributed systems:theory and practice. ACM Transactions on Computer Systems, 1992.
37. TCPA " Trusted Computing Platform Alliance", Building a trust in the PC, jan 2000, http://www.trustedcomputing.org(now known as trusted Computing Grpup, https://www.trustedcomputinggroup.org/home).
38. V. Varadharajan, "trust enhanced authorization and its applications", 2005.
39. B. Balacheff et al.," trusted computing Platforms: TCPA Technology in context", Prentice-Hall, 2003.Ramdomly---
40. M. Burrows, M. Abadi, R. Needham, a Logic of authentication, In: proceedings of the 12 th ACM symposium on Operating Systems Principles, 1989.
41. D. L. Hoffman, T. P. Novak, M. Peralta, Building consumer trust online, Communications of the ACM 1999.
42. Abdul-Rahman , A., Hailes,S.,: A Distributed Trust Model . In Proceedings , ACM New Security paradigmsWorkshop'97, Cumbria, UK 1997.
43. Wagealla, W., Carbone, M., English, C., Terzis, S., Nixon. P.: A formal model on trust lifecycle management. In : wksp on formal Aspects of security and trust (FAST 2003)at FM 2003. VOL IIT TR-10 / 2003. IIT-CNR, Itlay 2003.
44. Resnick, P., Zeckhauser, R.,Friedman, E., Kuwabara, K.: reputation Systems. Communication of the ACM ,2000.
45. R Yahalom,B Klein and T Beth. Trust relationships in secure systems-a didtributed authentication perspective.Proceedings of IEEE Conference on Research in Security and Privacy,1993.
46. B.Lampson,M. Abadi,M. Burrows, and E.Wobber.Authentication in distributed systems:Theory and practice.ACm Transcations on Compuetr Systems,1992,10(4),pp.265-310.
47. Matt Blaze,Joan Feigenbaum,and Jack Lacy.Decentralised trust management .In Proceedings of the 1996 IEEE conference on security and privacy,Oakland,CA may,1996,pp.164-173
48. C.Lin,V.Varadharajan,"Trust Enhanced Security-A New Philosophy for Secure Collaboration of MobileAgents"Proceedings of the Workshop on Trusted Collaboration,part of CollaborateCom2006,Atlanta,Georgia,USA.pp.17-20.
49. R. Yahalom, B. Klein ad T. Beth. Trust relationships in secure systems- a distributed authentication prospective. Proceedings of IEEE Conference on research in Security and Privacy, 1993.