Extending Trust Transitivity in Databases

Shatabdi Mondal
Asansol Engineering College,
Vivekananda Sarani, Asansol, Burdwan,
West Bengal – 713 305, INDIA,

S.K.Setua
University of Calcutta,
92, Acharya Prafulla Chandra Road,
Kolkata, West Bengal-700 009, INDIA,

ABSTRACT
Trust management now-a-days provides a promising approach for supporting access control in open environments. Trust transfer is a common technique employed in trust management systems to establish relationships between the two parties involved in a transaction who are strangers. Trust negotiation occurs when the two parties establish trust over the web. The paper proposes a model showing how trust can be transferred from a previous trust context to a new trust context.

General Terms
Database Security

Keywords
Trust, Reputation, Trust Transitivity, Trust Transitivity Context.

1. INTRODUCTION
Modern day databases are open, fine-grained, customizable and transparent [1]. They differ from traditional databases which were centralized [2]. One of the major objectives of trust management is to build up trust between two strangers or parties involved in database transactions. The parties involved in a transaction are known as trustor and trustee. A new trust can originate from an already existing old trust. The process of deriving new trust from the old trust is known as trust transitivity. The paper proposes a model showing how trust can be transferred from a previous trust context to a new trust context.

The paper is organized as follows. Section II provides a review to the already existing trust models, Section III defines the various forms of trust, Section IV proposes a model for trust transfer. Section V indicates the Future Work and Conclusion and Section V lists the References.

2. REVIEW WORK OF TRUST MODELS
In human society, trust depends on a host of factors which cannot be easily modeled in a computational system. Past experience with a person and with their friends, opinions of the actions a person has taken, psychological factors impacted by a lifetime of history and events (most completely unrelated to the person we are deciding to trust or not trust), rumor, influence by others' opinions, and motives to gain something extra by extending trusts are just a few of these factors. For trust to be used as a rating between people in social networks, the definition must be focused and simplified.

Marsh (1994) [3, 4] is among the pioneers to introduce a computational model for trust in the distributed artificial intelligence community. The model of reputation is absent in his work. Several limitations exist for his simple trust model. Firstly, trust is represented as a subjective real number between the range –1 and +1. The model exhibits problems at the extreme edges and at 0. Secondly, the operators and algebra for manipulating trust values are limited and have trouble dealing with negative trust values. Marsh also pointed to difficulties with the concept of "negative" trust and its propagation.

His model is complex and based on social and psychological factors. The model is highly theoretical and difficult to implement. It is particularly inappropriate for use in social networks because his focus was on interacting agents that could maintain information about history and observed behaviors. In social networks, users assign a single rating without explicit context or history to their neighbors and thus much of the information necessary for a system like Marsh's is missing.

Abdul-Rahman, et al, (2000) [3] have proposed that the trust concept can be divided into direct and recommender trust. These represent direct trust as one of four agent-specified values about another agent ("very trustworthy", "trustworthy", "untrustworthy", and "very untrustworthy"). Here, recommended trust is known as "reputation". The translation from recommendations to trust is performed through an ad-hoc scheme. Ad-hoc formulation plagues several other proposals for reputation/trust systems such as those in Glass, et al. (2000), Yu and Singh (2001), Esfandiari, et al., (2001), Rouchier, et al. (2001), Sabater, and et al., (2001), among others. Nevertheless, reputation and trust have been found to provide useful intuition or services for of these systems.

Lik Mui, Mojdeh Mohtashemi, Ari Halberstadt (2002) [3] have proposed the computational model of trust. Reputation, Reciprocity, Encounter, History are discussed in this model.

Here a new parameter was introduced i.e. History to get previous information of transaction.

3. DEFINITIONS
Alternative definitions of Trust: Kini and Choobineh [5] have defined trust as "Trust in a system is defined as an individuals belief in the competence, dependability, and security of the system under conditions of risk." Kini and Choobineh[5], state that trust, as defined in the Webster dictionary, is:

- An assumed reliance on some person or thing. A confident dependence on the character, ability, strength, or truth of someone or something.
- A charge or duty imposed in faith or confidence or as a condition of a relationship.
- To place confidence (in an entity).

The Oxford Reference Dictionary has stated that trust is "the firm belief in the reliability or truth or strength of an entity."

The European Commission Joint Research Centre has defined trust as "the property of a business relationship, such that reliance can be placed on the business partners and the business transactions developed with them" [6,11].

To be more specific, trust does not mean the following:-First, trust is not reliance on some other person. We may have to rely on some other person for something but that does not mean we trust him. Second trust is not jurisdiction. If one's system manager tells that he has got 200 KB of disk space reserved. One will have to trust him but if the system manager tells that the system manager will extend it to 500KB tomorrow one may not trust him. Third, trust is not delegation. When we delegate an authority over X to somebody that does not necessarily mean that we trust him for that purpose [7].

Thus, the definition of trust incorporates two forms of trust-reliability trust and decision trust.

Reliability Trust: Reliability trust is a subjective trust [8] which differs from one user to another user in a particular group where members are reliable to each other. The data that is worked upon must be reliable because we neither have the past data nor the future data. This means the past data must not reflect that a particular user is not capable of working upon the present data and at the same time we cannot predict the future data. The introduction of mechanisms for referential integrity through the use of foreign key has required the introduction of a related access mode allowing a user to refer a table from another table. This very mechanism depends on the base table from which referential integrity is being made, thus incorporating the essence of reliability trust.

Decision Trust: This kind of trust incorporates a notion of relative security [8]. Such trust is required for a user working in a group and the group is entrusted with an access control list. Another notion of relative trust is incorporated when a user works with a particular dataset. A user fetches data from the database by database queries. The set of legal insertions and updates is constrained to those that do not create two entities with the same value on a candidate key.

Literature survey explores two more forms of trust-Credential-based trust and Reputation-based trust.

Credential-based trust: Credentials are digitally signed assertion [4]. A credential can have multiple usages. Firstly, delegation of attribute authority:-an entity delegates the authority over an attribute to another entity by the grant command. Secondly, inference of attributes: - an entity uses one attribute to make inference about another attribute. Thirdly, attribute fields:-attribute credentials carry field values such as ssn number and balance limit. These fields are used to infer additional attributes based on these field values. They are also used to delegate attribute authority to a certain entity only for certain specific field values. Finally, attribute-based delegation of attribute authority-with this strangers' trustworthiness is determined based on their certified attributes [9].We can thus use the credentials for role activation and to make selective use of roles and the evaluation of access control.

Reputation based trust: Reputation can be considered as a collective measure of trustworthiness based on the referrals or ratings from members in a community or group [4].

A multilevel secure database contains different access classes. Users are in different access classes.

A user in a group has a particular access control. An access class dominates if firstly, users of that class contain higher access right, and secondly, if there is more number of users in an access class then that access class dominates over other access classes.

A group with higher access class has higher reputation and vice-versa. The security level of the access class associated with data reflects the sensitivity of the information contained. That is, on the other hand, to say, the potential damage that could result from the unauthorized disclosure of the contents of the data.

An authorization can be granted to all members of a group, and a specific authorization is granted for a specific member. Thus the group might be granted a positive authorization and the specific member the negative authorization. In this case, the authorizations granted directly to the user are more specific than the authorizations granted to the groups to which the user is a member. Authorization directly granted to user take precedence over authorizations specified for groups to which the user belongs and null mode authorization given to a user overrides any other authorization granted to the same user. Thus negative authorization always overrides positive authorization [10].

When a database operation occurs, it embodies some kind of trust between the user and the database operation performed. This paper proposes different trust terminologies according to the database operations.

Insertion Trust: When we refer a foreign key we are to check whether the corresponding primary key is there or not. At the same time when we insert a primary key we must check that the primary key is not a null value. This is known as insertion trust where we trust that the primary key already existed in the system and it is not null.

Deletion Trust: Before the deletion of a tuple, we are to ensure that all references created by the tuple directly or indirectly through foreign keys is deleted. This is known as deletion trust.

Updation Trust: Setting new values to a particular attribute of the database should be made taking into granted all the other attributes that the attribute directly or indirectly affects like the aggregate functions, which should also be updated accordingly. This is known as updation trust.

Selection Trust: Trust is incorporated when selection of tuples is done. Trust lies in the fact that there is no duplicity of tuples in the resulting relation.

Projection Trust: Trust is incorporated when we project tuples. The projection operator deletes all other tuples that are not in the projection list but are present in the table. This is projection trust where we trust projection operator will take up only that attribute that is supposed to project.

Join Trust: Trust is incorporated in the join condition itself where join performs a cross product of two tables and returns only those variables related to the join. Thus, join trust embodies trust in the join operator.

Transaction is a logical unit of work. According to the transaction and its various features and functionalities we have incorporated some trust definitions.

Transaction Trust: Transaction is a logical unit of work. This means a particular unit of work completes entirely or does not happen at all. The commit operation shows successful end-of transaction. It indicates that a logical unit of work has been successfully completed and all the work done by the transaction has been saved or made permanent. The trust incorporated in committing the data to a savepoint is known as commit trust and, if the transaction is not successful then the work done after the previous transaction was committed, must be rolled back. The trust associated when the data is taken back to the previous transaction is known as rollback trust and the two together comprise the transaction trust.

Consistency Trust: Transaction is a logical unit of work. This means a particular unit of work completes entirely or does not happen at all. In either case, the database is in a consistent state. This means transaction will render the database a safe state. The trust incorporated in a transaction where a database is safe before a transaction and again regains safety after a transaction is known as consistency trust.

Concurrency Trust: Concurrency allows many transactions to access the same database at the same time. To allow this, when one transaction is updating the value of an attribute then no other transaction will update the value of the same attribute. To attain this, attributes attain locks. The trust incorporated where attributes attain locks to maintain concurrency is known as concurrency trust.

Serializability Trust: Two equivalent schedules produce the same result independent of the initial state of the database. The trust incorporated that the two schedules will produce the same result is known as serializable trust.

Transient Trust: A view table is a temporary file generated from the logical level data and the physical level data. The trust incorporated in generating data at the view level data from data at the physical level and logical level is known as transient trust.

Trust Transitivity

Definition of Trust Transitivity: Trust is not always transitive. However, there are some factors associated with trust which can make trust transitive. Firstly, the purpose of the trust or the context or situation under which A trusts B and B trusts C which makes A trust C. Secondly, a trust measure or the degree of trust associated with the trustor and the trustee. Examples of trust measure are strong trust, weak trust, strong distrust and weak distrust. Lastly, time is an important factor for trust. One might trust someone today but one might not trust him after some day or after some transactions.

The access control policies are used in databases for trust transitivity. This means in a multilevel secure (MLS) database, the user at the highest access level can transfer the access right to some user at the next authentication level and so on by the grant predicate.

In context of the database operations, this paper proposes definitions on transitivity of trust.

Insertion Trust Transitivity: From the definition of insertion trust we can infer that trust gets transmitted from an attribute to a key attribute and from the key attribute to the referential attribute.

Attributes ⟶ Key Attribute ⟶ Referential Attribute.

Deletion Trust Transitivity: From the definition of deletion trust we can infer that trust gets transmitted from an attribute to the referential attribute and from the referential attribute to the key attribute.

Attributes ⟶ Referential Attribute. ⟶ Key Attribute

Updation Trust Transitivity: From the definition of updation trust we can infer that trust gets transmitted from the updated (new) value to the updated value of the results computing aggregate functions (U1) and from the values of aggregate functions to the value which are either output or are used as input for fetching still some other results (U2).

Updated values ⟶ Updated values in agg. functions (U1) ⟶ Values used again as input to some other value (U2).

Selection Trust Transitivity: From the definition of selection (σ) trust we can infer that trust gets transmitted from the input tables to the resulting table(S_1) after selection operator has been used and from the tuples obtained after selection operator is used to the tables affecting values of tuples obtained after the selection operator has been used(S_2).

Input tables ⟶ S_1 ⟶ S_2.

Projection Trust Transitivity: From the definition of projection(Π) trust we can infer that trust gets transmitted from the tuples in the projection list to the tuples obtained after projection operator has been used(P_1) to the tables using the result obtained after the projection operator has been used(P_2).

Input tuples ⟶ P_1 ⟶ P_2.

Join Trust Transitivity: From the definition of join(⋈) trust we can infer that trust gets transmitted from the tables to the values obtained after the join operation(J_1) has been performed to the values affected by using the values obtained by the join operator(J_2).

Input tuples ⟶ J_1 ⟶ J_2.

According to the features and its various functionalities of transaction, transitivity of trust is defined.

Transaction Trust Transitivity: From the definition of transaction trust we can infer that trust is incorporated not only in one transaction but also in individual sub-transactions. If the transaction occurs, the trust gets transmitted from these individual sub-transactions to transactions and from the transactions to the savepoint. This is known as commit trust transitivity. In case, the transaction fails to occur a rollback occurs. The transitivity of trust from the savepoint to the transaction and from the transaction to the individual sub-transactions is known as rollback trust transitivity.

Consistency Trust Transitivity: From the definition of consistency trust we can infer that trust gets transmitted from the sub-transactions to the transactions before a commit operation occurs and from the point at which the commit operation occurs to the transactions and from the transactions to the individual sub-transactions.

Concurrency Trust Transitivity: From the definition of concurrency trust we can infer that trust gets transmitted from the state of the sub-transaction to the state of the whole transaction and from the state of the whole transaction to the state at which the attributes attain locks.

Serializability Trust Transitivity: From the definition of serializability trust we can infer that trust gets transmitted from the schedule to the transaction and from the transaction to the sub-transactions.

Transient Trust Transitivity

Trust gets transmitted when generating view level data. Trust transitivity occurs from the physical level data to the logical level data and from the logical level data to the view level data. Such trust is known as transient trust.

The transient trust can be used for transitivity of data. The view table generated at the highest access class can be used to generate data set, which by careful SQL injections can lead to another view table which when again operated will give the proper results Thus, the view table is used for trust transitivity. The group can give the access right to some group and the second group can give the access right to a third group. Similarly, for a sub-group the first sub-group transfers trust to a second sub-group and the second sub-group and so on.

4. PROPOSED TRUST MODEL

Now we are going to propose a trust model. Each database operation can be assumed to be a context which can be of Insertion Trust Transitivity Context, Deletion Trust Transitivity Context, Updation Trust Transitivity Context, Selection Trust Transitivity Context, Projection Trust Transitivity Context and Join Trust Transitivity Context.

Insertion Trust Transitivity Context: Trust gets transmitted from an attribute to a primary key and from the primary key to the foreign key. Here current node is primary key and next node is foreign key.

Deletion Trust Transitivity Context: Trust gets transmitted from an attribute to the referential attribute and from the referential attribute to the key. Here current node is foreign key and next node is key attribute.

Updation Trust Transitivity Context: Trust gets transmitted from the updated (new) value to the updated value of the results computing aggregate functions and from the values of aggregate functions to the value which are either output or are used as input for fetching still some other results.

Selection Trust Transitivity Context: Trust gets transmitted from the input tables to the resulting table after selection operator is used and from the tuples obtained after selection operator is used to the tables affecting values of tuples obtained after the selection operator is used.

Projection Trust Transitivity Context: Trust gets transmitted from the tuples in the projection list to the tuples obtained after projection operator has been used to the tables using the result obtained after the projection operator is used.

Join Trust Transitivity Context: Trust gets transmitted from the tables to the values obtained after the join operation is performed to the values affected by using the values obtained by the join operator.

Implementation:

For a Context C1.

1. Each node maintains a database or a table which consists of three fields: Context ,Next Pointer and Previous Pointer.

Context(C1)	Next Addr	Previous Addr

2. Let, Node A wants to transfer trust to Node B
   i) First it will check if there is any path to B by Traversing Double Linked List.
   If it gets more than one path then it will choose the Shortest path by using any Shortest path algorithm.
   Else if there is no path then add another tuple which consists of context c1 and Next pointer address which will address the destination node. Now, Table of destination node will be updated by adding another tuple which consists of context c1 and previous pointer address which will address the Source node as shown in Fig.1.

   Example:

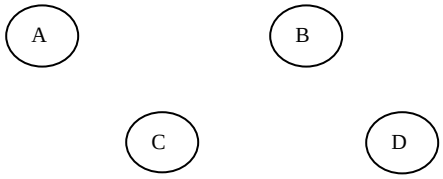

Fig.1

Let, A wants to transfer trust to B as shown in Fig.2.

First it will find any path. No path is there so there will be a direct connection.

Database of A

Context	Next	Previous
Context(C1)	B	Null

Database of B

Context	Next	Previous
Context(C1)	Null	A

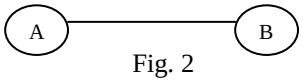

Fig. 2

Now, A wants to Communicate

First Control search a address of next pointer from table then it will go to B then Next fields of B is Null so it can not construct a path. So, there will be a direct connection.

Database of A

Context	Next	Previous
Context(C1)	B	Null
Context(C1)	C	Null

Database of C

Context	Next	Previous
Context(C1)	Null	A

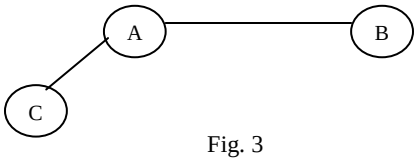

Fig. 3

Now, B wants to communicate to C as shown in Fig. 3
B first finds is there any path to C . It will get B-A-C.
Now, B wants to communicate to D
B first finds is there any path to D. There is no path.
So, there will be a direct connection as shown in Fig. 4.

Database of B

Context	Next	Previous
Context(C1)	D	A

Null Value will be updated by D.

Database of D

Context	Next	Previous
Context(C1)	Null	B

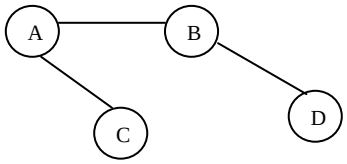

Fig. 4

The model only focuses on the way trust gets transferred.

5. FUTURE WORK AND CONCLUSION

The paper focuses on trust, explains the various forms of trust and lastly proposes a model for trust transfer. Our future work comprises of proposing a model for the delegation of trust and calculating the trust value for various types of transactions.

6. ACKNOWLEDGMENTS

We express our sincere thanks to the all the members of the Departments of Computer Science and Engineering of Asansol Engineering College and Rashbehari Siksha Prangan without whose help the development of the paper would not have been possible.

7. REFERENCES

[1] Paul Alan Porter, "Trust Negotiation for Open Database Access Control" A thesis submitted to the faculty of Brigham Young University in partial fulfillment of the requirements for the degree of Master of Science Department of Computer Science Brigham Young University August 2006.

[2] Bertino Elisa, Sandhu Ravi "Database security - Concepts, Approaches and challenges", IEEE Transactions on Dependable and Secure Computing, vol.2, no. 1, pp. 2-19,2005

[3] Lik Mui Mojdeh Mohtashemi, 200 Technology Square, Cambridge, MA 02139, USA {lmui, mojdeh}@lcs.mit.edu, Ari Halberstadt 9 Whittemore Road,Newton, MA 02458, USA *ari@magiccookie.com*, "A Computational Model of Trust and Reputation." Proceedings of the 35th Hawaii International Conference on System Sciences – 2002, pp 2431 - 2439

[4] Donovan Artz and Yolanda Gil,Information Sciences Institute University of Southern California 4676 Admiralty Way, Marina del Rey CA 90292 +1 310-448-9197, +1 310-822-1511 {dono,gil } @isi.edu February 8, 2006, "A Survey of Trust in Computer Science and the Semantic Web", Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 5, and No. 2. (June 2007), pp.58- 71.

[5] A. Kini and J. Choobineh, "Trust in Electronic Commerce: Definition and Theoretical Considerations," *31st Annual Hawaii Int'l. Conf. System Sciences*, 1998, Hawaii, http://ieeexplore.ieee.org/iel4/5217/14270/00655251.pdf

[6] S. Jones, "TRUST-EC: Requirements for Trust and Confidence in E-Commerce," 1999, European Commission, Joint Research Centre.

[7] Bruce Christianson and William S. Harbison, "Why Isn't Trust Transitive?" In Proceedings of the International Workshop on Security Protocols (1997), pp. 171-176.

[8] Audun Jøsang, Information Security Research CentreQueensland University of Technology Brisbane, Australia, Roslan Ismail, College of Information Technology Universiti Tenaga Nasional (UNITEN) Malaysia, Colin Boyd, Information Security Research Centre Queensland University of Technology Brisbane, Australia, "A Survey of Trust and Reputation Systems for Online Service Provision." Preprint of article published in Decision Support Systems, 43(2) 2007, p.618-644

[9] Ninghui Li, John C. Mitchell, William H. Winsborough, "Design of a Role-based Trust-management Framework." IEEE Symposium on Security and Privacy, 2002. Proceedings, pp-114- 130.

[10] T.F. Lunt, D.E. Denning, R.R. Schell, M. Heckman and W.R. Shockley, "The SeaView security model." *IEEE* Transactzons on Softtoare Enganeerzng, vol. 16, pp. 593-607, 1991.

[11] Tyrone Grandison and Morris Sloman, Imperial College, "A Survey of Trust In Internet Apllications", IEEE Communications Surveys, http://www.comsoc.org/pubs/ surveys Fourth Quarter 2000.