

TRIUMF: A Trusted Middleware for Fault-tolerant Secure Collaborative Computing

Waseem Ahmad *, Ashfaq Khokhar†

*Department of Electrical and Computer Engineering

University Of Illinois at Chicago

Chicago, IL 60607

Email: wahmad@ece.uic.edu

†Department of Electrical and Computer Engineering

University Of Illinois at Chicago

Chicago, IL 60607

Email: ashfaq@ece.uic.edu

(Working Draft)

11.01.2005

Abstract

A collaboration is an activity conducted by two or more parties to achieve a common goal. Business collaborations are becoming an essential part of emerging business models. Organizations, however, are unable to reap true benefits of collaborations because of their security and privacy concerns. TRIUMF, the Trusted Middleware for Fault-tolerant secure collaborative computing, is aimed at enabling privacy preserving collaborative processes across administrative domains. The middleware is built around a Services Oriented Architecture comprising of an ensemble of services. The membership services are responsible for secure and authenticated entry into the collaboration process. Data Request Processing Services are responsible for deciding the level of privacy required along with the selection of appropriate aggregation functions and/or proper Secure Multiparty Protocols. Secure Data Access Services are responsible for providing efficient access to private and non-private data sources. Privacy Preserving Data Processing services provide mechanisms whereby secure sharing of data, information or knowledge is made possible. The High Availability services provide mechanisms to ensure fault tolerant execution of applications running on TRIUMF. TRIUMF is being developed on top of JXTA, which provides an open source P2P networking infrastructure.

I. INTRODUCTION

A collaboration is an activity conducted by multiple parties to achieve a common goal. Emerging business models require organizations to collaborate with each other [2]. Collaborations are seen in many areas whether they are

TRIUMF middleware project is supported by an NSF SGER grant

in the form of outsourcing, whereby one organization hires another organization's services to complete a task, or in the form of law enforcement agencies sharing information with each other to identify and locate potential criminals or health related agencies exchanging patient-disease patterns to provide better health care facilities. These collaborations, however, are mostly limited in scope because of the privacy and security concerns of the individual organizations. These concerns are reflected in organizational security policies and are effected by certain federal or state regulations. The examples of such regulations are Health Insurance Portability and Accountability Act (HIPPA) in the field of health care, Gramm-Leach-Bliley Act (GLBA) in financial services, California's Database Security Breach Notification Act SB 1386, Europe's Data Privacy Act and Canada's Personal Information Protection and Electronic Document Act(PIPEDA).

TRIUMF stands for Trusted Middleware for Fault Tolerant Collaborative Computing and is aimed at helping organizations reap true benefits of collaboration while making sure that there are no compromises made to their security and privacy policies. In TRIUMF, collaborations are seen as sharing of (raw)data, information and/or knowledge. Information can be viewed as statistical aggregates like sum, average, median etc taken over raw data. The knowledge on the other hand can be viewed as the end product of some data mining operation intended for knowledge discovery. TRIUMF takes a novel approach to the privacy preserving collaboration problem by providing security guarantees on various dimensions. For example, consider the example scenario of Center for Disease Control (CDC) interested in monitoring the spread of a specific disease in a certain area. For that purpose, it needs to access patient records from all the hospitals in the area. Considering the private nature of the patient records, no hospital will be able to share the individual medical records due to its internal privacy policy which in turn is effected by federal and state regulations (some of them are mentioned above). Hospitals can, however, share aggregates over the entire patient records, such as percentage of the patients suffering from a specific disease, without having to release the identity of any individual patient. But even sharing the aggregates run into problems as, for example, no hospital is interested in releasing its mortality rate to either of CDC or any other hospital. Similarly knowledge gleaned from patient records can help reveal vital statistics without revealing individual patient identities. There are situations where sharing of knowledge is more desirable than just the aggregate. Like for example a Health Insurance Provider would be more interested in knowing what kind of personal attributes in an individual make him susceptible to a specific disease. These kinds of queries can be answered in the form of association rules such as "*A person who has been smoking from the last ten years is more likely to develop lung cancer*". This, however, does not mean that all such associations can be revealed to every one. For example any association rule which involves SSNs, Age and Zip codes together can lead to privacy violations. For example an association rule of the form "*Persons aging more than X and living in zip code Y have a disease Z with confidence 100%*" when matched with the publically available records can lead to privacy compromise for the individuals aging more than X in the zip code Y. TRIUMF tackles these problems by providing a policy driven approach to data access, information aggregation and knowledge discovery with subsequent sharing by allowing organizations to describe their privacy and security policies in a very fine grained manner. For example, in the above scenario if an organization considers an aggregation over a specific attribute to be private then its security policy will reflect it and hence TRIUMF would not allow such aggregations

to be shared with anyone outside the organization. Similarly if the security policy dictates that certain attributes and combinations of them are private and that they can't appear in knowledge shared with other organizations then TRIUMF would suppress such attributes so that there is no privacy violation. There are other situations as identified by Sweeney [14] where organizations themselves are not fully aware of the risks involved in releasing information. The approach suggested in [14] is to generalize the data such that its impossible to track an individual from $k - 1$ individuals, this approach was named as $k - anonymity$. We believe this approach can be useful in collaborative processes as well, therefore TRIUMF is being designed to make use of this approach wherever required.

To get the intended job done, TRIUMF provides five categories of services. The membership services allow organizations to join new collaborative processes. These services also allow them to dictate who they want to collaborate with and what kind of credentials are required by any newcomer who wants to join the collaborative process. The data access services allow an organization to define privacy level on each and every data source by allowing it to declare a data source as entirely or partially private (some attributes and their combinations as private and others as non-private). The privacy preserving services allow secure and private computation of collaborative functions by providing efficient realizations of novel secure multiparty computation protocols. High Availability services allow collaborative applications to efficiently replicate themselves (when enough resources are available) or do periodic checkpointing in order to enable high availability applications. These services also provide a fault tolerant messaging framework.

Rest of the paper is organized as follows. Section *II* provides an introduction to the TRIUMF's service oriented architecture along with the detailed explanation of individual services. Section *III* describes TRIUMF's Privacy Preserving Data Processing services in detail. Section *IV* provides some insight into the implementation of TRIUMF. Section *V* highlights related work and Section *VI* provides concluding remarks.

II. TRIUMF ARCHITECTURE

A high level architecture for TRIUMF is shown in figure 1. TRIUMF nodes are shown connected through internet (it can be a private network as well). TRIUMF essentially creates a Virtual Private Network even when underlying network infrastructure is public internet. The TRIUMF nodes act as gateway nodes as they connect internal organizational resources to the TRIUMF network. There is a certificate server shown in the figure (there can be multiple of these depending upon the performance and fault tolerance requirements). Nodes get their necessary credentials from certificate server.

A. TRIUMF's Services

Figure 2 shows the set of services provided by TRIUMF. A detailed explanation of these services is as follows.

- 1 **Membership Services:** The membership services allow a TRIUMF node to join new collaborations by providing relevant credentials obtained from TRIUMF certificate server. These services are also responsible

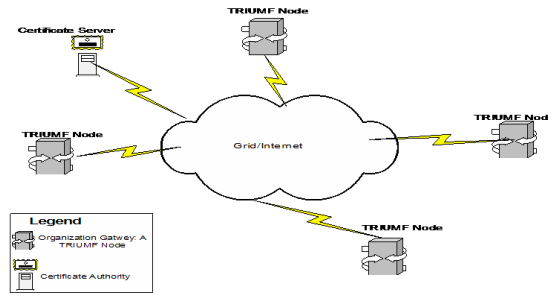


Fig. 1. TRIUMF's High Level Architecture

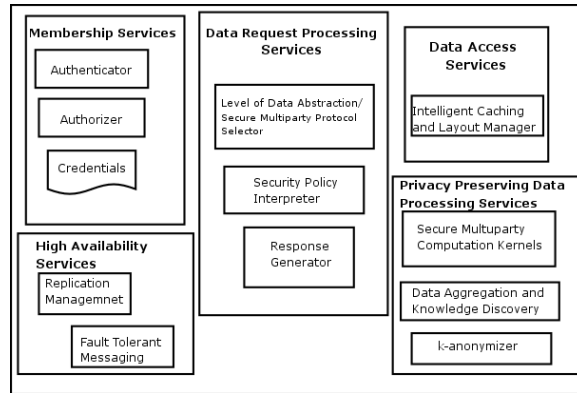


Fig. 2. TRIUMF's Service Oriented Architecture

for allowing existing members of a collaborative group to dictate who is allowed to join and who is not and also what trust levels are assigned to the newcomers.

2) **Data Request Processing Services:** In an active collaboration, a node will have to cater for the data requests coming from other participating nodes. Data Request Processing services allow any TRIUMF node to perform following tasks.

- 1) To assign each incoming request a set of rights based on the trust level on the requester.
- 2) To determine what level of data abstraction and/or what kind of secure multiparty computation is required to meet the request based on the associated rights under privacy constraints.

The request authorization is done based on Role Based Access Control(RBAC). As explained in the introduction part, efficient realization of secure information sharing requires that intelligent decisions be made as to what level of aggregation would suffice to ensure privacy. There are some situations, however, where simple aggregates can no longer solve the purpose. Like for example simple aggregates would not suffice in the case where CDC asks area hospitals to determine what are the specific patterns behind the spread of a disease over the entire area. The problem however can be solved through Distributed Data Mining based on Secure Multiparty Computation(SMC) kernels (popularly called as Privacy Preserving

- Data Mining). The identification of required SMC kernels is the job of Data Request Processing Services.
- 3 **Data Access Policy Service:** This service allows organizations to describe their security and privacy constraints in a dynamically updateable fashion. It allows fine grained enforcement of security policies by allowing organizations to define a set of private and non private data sets for each trust level such that organizations are able to indicate a combination of private attribute set as well in case of private data sets.
 - 4 **Data Access Services:** These services are responsible for laying out and caching data in Organizational Data stores for efficient data access. The data layout and caching plans take into consideration the private/non-private nature of the data sets.
 - 5 **Secure and Privacy Preserving Data Processing Services:** The services are pivotal to the overall functionality of the TRIUMF middleware. These services provide a variety of data aggregation and knowledge discovery functions in both Cryptographic Secure Multiparty Computation and Non-Cryptographic (K-anonymity, Randomization) settings. We will provide a detailed explanation of these services in the next section.
 - 6 **High Availability Services:** These services allow TRIUMF applications to replicate themselves when enough resources are available in order to meet high availability requirements in case of system failures on certain nodes. This would mean that an organization may have multiple gateway nodes which connect it to the TRIUMF network. When replication cost becomes high, these services allow applications to perform periodic checkpointing instead of replication so as to be able to roll back to a consistent state after a system failure.

III. PRIVACY PRESERVING DATA PROCESSING SERVICES

Privacy Preserving Data Processing Services provide various forms of aggregation and data mining services under Cryptographic Secure Multiparty Computation and non-Cryptographic yet privacy preserving settings. Below is the detailed account of both settings.

A. Cryptography Based Secure Multiparty Computation(SMC) Approach

Under SMC setting, multiple parties collectively compute a common function of their local inputs such that no party learns more than what can be learnt from their own inputs and the result of the function. It was first proposed by Yao [10] for two party case. Later Goldreich et al [11] extended this approach to multiple parties and provided theoretical proofs to show that secure protocols exist for any function. Such general protocols, however, require all-to-all communication operations rendering them extremely inefficient for large scale networks. Benny Pinkas et al [5] applied secure multiparty computation protocols to solve privacy preserving data mining (Decision Tree Induction) problem. Their solution replaces each data exchange operation in ordinary data mining algorithms with a cryptographic primitive which provides the same result without disclosing the data of the participants. This design uses synchronized all-to-all communication patterns and requires re-computation if the data changes. This results in poor scalability.

Kantarcioglu, Vadiya and Clifton [9] extended Pinkas et al's approach to other data mining algorithms such as Clustering and Association Rule Mining. Their protocols thus suffer from the same inherent limitations. Gilburd, Schuster and Wolf [7] proposed SMC protocols in context of privacy preserving association rule mining problem which can support large number of participants. They however relaxed the privacy requirement to $k - TTP$ (a participant is able to know results over k participants where $k \ll n$, the total number of participants). The most differentiating aspect of their protocol was the use of additively homomorphic cryptosystem.

Most of the earlier work in applying Secure Multiparty Computation ideas to privacy preserving data sharing problem assume semi-honest adversarial model. It is widely being felt that there is a need for protocols which are designed with malicious adversaries in mind as we move towards large scale overlay networks built on top of public internet.

B. Additively Homomorphic CryptoSystems

Using Additively Homomorphic Cryptosystem, one can combine ciphertexts into a new ciphertext that is the encryption of the sum of the messages of the original ciphertexts. Paillier [12] proposed an efficient Additively Homomorphic Cryptosystem. Jurik et al [13] provided length flexible and threshold versions of Pailliers Cryptosystem. A threshold scheme is a method of sharing a message M among a set of w participants such that any subset consisting of t participants can reconstruct the message M but no subset smaller than t can reconstruct M .

TRIUMF makes use of the threshold, length flexible cryptosystem proposed by Jurik et al [13]. Using that cryptosystem several multiparty protocols are provided as part of TRIUMF's crypto and communication libraries. Some of these protocols are as follows.

- Highly Scalable Secure Aggregation protocol over asynchronous ad-hoc networks.
- A mix-net, which is a tool for making an unknown random permutation of a list of ciphertext. This makes it useful for achieving anonymity.
- A key escrow system, which allows an authority to decrypt any message sent using any public key set up in the default way.

1) *TRIUMF's Highly Scalable Secure Aggregation Algorithm*: The proposed algorithm employs gossip based broadcast primitives for data dissemination and aggregation. Gossip based algorithms although probabilistic in nature, have been shown to have excellent scale-up properties [15].

There has been some work done in gossip based secure aggregation protocols [16] but that was limited to sensor networks. Moreover, the algorithms presented heavily relied on the underlying network topology. In fact those algorithms assumed that the sensor nodes are laid out in a Grid which is not true in a lot of practical cases such as battlefield and environment monitoring applications. In sensor networks, resource constraints don't allow Public Key based Cryptographic functions. But there is no such constraint on TRIUMF nodes in B2B collaborations which we are focussing on at the moment.

The proposed algorithm does not allow any single node to decrypt the aggregates taken over any set of participants.

This makes sure that at no time during the execution of the algorithm, does any participant is able to decrypt partial aggregates taken over subset of nodes. This particular property is possible because of the threshold nature of the employed cryptosystem. This scheme is in line with the practical scenarios involving large number of nodes where multiple aggregation points are defined so as to remove any possibility of a single node being the bottle-neck. In our above mentioned example scenario where CDC is involved in collaborating with hospitals and HMOs in a certain area. The scale of the network would become clear if we note that this collaboration process might be spanned over the entire country. In that case, regional CDC centers would be collecting data from the nodes in the corresponding regions. In our scheme no single CDC regional center is able to determine the aggregation results of the data received from its region. Only when all the regional nodes collaborate, they will be able to determine the final aggregate in decrypted form but by this time the aggregation results employ values from all the nodes. This way perfect privacy is achieved instead of k -privacy as proposed in [7].

The algorithm starts with the selection of local hub nodes (regional CDC centers in above example) if they are not pre-defined. Note that in most of the practical applications, these hub nodes will certainly be pre-defined like in the above CDC example. However, if these hub nodes are not already defined, each node sends a message to each of its neighbors and then counts the total number of messages that it has received itself. This count value is then sent to all the nodes in the neighbor list. Each node on obtaining the counts associated with each of its neighbors, selects one of them (including itself) as hub whose count (i.e. the number of neighbors) is the highest. Once the hub nodes have been defined, they engage in a distributed key generation process using the additively homomorphic threshold cryptosystem. This key generation process will leave each hub node with a share of the private key along with the complete public key. Each hub node broadcasts this public key to all its neighbors. The epidemic broadcast would result in all the nodes in the network getting hold of the public key. All the nodes then engage in epidemic broadcast of the encrypted values such that at first step every node forwards the encrypted version (cipher-text) of its local value(plain-text) and during the second step, it will forward the result of the homomorphic operation on the cipher-texts corresponding to its local value and the values it received from its neighbors. Note that this function is performed on the cipher-texts and reveals nothing about the underlying plain-texts. After that, each node will forward the results of the same operation on the previous value forwarded and the new values received. Note that each node engages in the broadcast operation only when there is a change in either of its local value or the received values. After a certain time (which has been proved to be bounded by the log of the total number of nodes [17]), each node including the hub nodes will have the the cipher-text corresponding to the aggregate over the entire network. The hub nodes will then perform threshold decryption of this value. The threshold decryption process requires that at least t of the hub nodes are honest amid Byzantine adversarial conditions. Only when the required number of hub nodes perform decryption using their local share of the private key, then they will be able to have the actual value aggregated over the entire network. Note that we have focused on the decryption process to make sure that on one hand no node can get the aggregate value over a subset of the nodes in the network. And on the other hand, we have also made sure that malicious nodes in the network behaving under Byzantine model are unable to compromise the security of the protocol. The robustness of the aggregate operation against malicious

values can be ensured by techniques proposed by Wagner [18].

C. Non-Cryptographic Approach

The Secure Data Processing is not restricted to Cryptographic Secure Multiparty Computation applications. Secure Multiparty Computations although are very good at achieving privacy during collaborative processing. They are nevertheless quite expensive both in terms of communication and computation requirements. It is therefore highly recommended that SMC techniques be used intelligently only when absolutely necessary. As we have seen in the above discussion, we can achieve high degrees of privacy just by data abstraction through aggregation and knowledge discovery. This is the idea pursued in Non-Cryptographic approach. Based on the input from Policy service and Data Request Processing Services, one of Cryptographic SMC or Non-Cryptographic approaches are selected for secure data processing. Following functions are provided as part of Non-Cryptographic services.

- Distributed Data Aggregation Functions.
- Privacy Preserving Data Integration and subsequent OLAP (using randomization and/or k-anonymity).

IV. TRIUMF'S COMMUNICATION SERVICES

One of the design goals behind TRIUMF was to achieve excellent scale-up properties with a potential to scale upto millions of participants. The underlying communication framework, therefore, is peer to peer (P2P) which has been proven by Napster and Gnutella etc to support extremely large scale collaborations. TRIUMF is aimed at making use of JXTA and Globus combined with the support for Web Services. TRIUMF's communication library provides an abstraction to the underlying network services. Our initial focus, however, is on JXTA support.

A. Overview of JXTA

- JXTA technology is a set of simple, open peer-to-peer protocols that enable any device on the network to communicate, collaborate, and share resources.
- JXTA peers create a virtual, ad hoc network on top of existing networks, hiding their underlying complexity. In the JXTA virtual network, any peer can interact with other peers, regardless of location, type of device, or operating environment even when some peers and resources are located behind firewalls or are on different network transports.
- JXTA technology runs on any device, including cell phones, PDAs, two-way pagers, electronic sensors, desktop computers, and servers.
- Based on proven technologies and standards such as HTTP, TCP/IP and XML, JXTA technology is not dependent on any particular programming language, networking platform, or system platform and can work with any combination of these
- The JXTA addressing model is based on a uniform and location independent logical addressing model. Every network resource (peer, pipe, data, peergroup, etc.) is assigned a unique JXTA ID.
- All network resources in the JXTA network, are represented by advertisements.

```

Algorithm SS_Aggregate

do forever

/*Used in cases where Hub Nodes are not already defined*/
HubNode = selectHNode();
/*Check if I am the Hub Node*/
if(me == HubNode)
    *Find out who are other Hub Nodes*/
    HNList = findHubNodes();
    *Generate Public-Private Key pair using a distributed protocol employing all other hub nodes. The function DKG(Distributed Key
    Generation) leaves every hub node with the entire Public Key and a share of the private key.*
    (PublicKey, PrivateKey) = DKG(HNList);
    /*Use epidemic broadcast function to propagate Public Key*/
    broadcast(PublicKey);
endif
/* If I am not a Hub Node, I will listen to Broadcast channel for Public Key*/
listenToBroadCastChannel(pkChannel, publicKey);
/*Encrypt my local value with the received public key and broadcast it to neighbors*/
broadcast(Enc(PublicKey, LocalValue));
/*Listen to the Broadcast Channel for Ciphertext Values from neighbors*/
listenToBroadCastChannel(AggregateChannel,
ReceivedCipherTexts);
/*Combine the Recieved Ciphertexts with local Ciphertext*/
combinedCipher = combineCipherTexts(LocalCipherText, ReceivedCipherTexts);
/*Broadcast the combined cipher texts on the broadcast channel, if there is a change in the local value or if there is a new value received.*
if(LocalValueChanged == true||NewRecieved == true)
broadcast(combinedCipher);
/*Okay, we have all the values now. Lets decrypt the entire combined ciphertext*/
if(me == HubNode)
aggregate = thresholdDecrypt(combinedCipher, HNList)

endAlgorithmSS_Aggregate

```

TABLE I

TRIUMF'S EPIDEMIC BROADCAST BASED SECURE AGGREGATION PROTOCOL

```

Algorithm SelectHNode
/*broadcast to the neighbors indicating my existence as their neighbor*/
broadcast(1);
/*Listen to the default broadcast channel*/
listenToBroadCastChannel(defaultChannel,numNeighbors);
broadcast(numNeighbors);
listenToBroadCastChannel(defaultChannel,NeighborCountPair[]);
hubNode = NeighborCountPair[maxIndex].getNeighborID();
returnhubNode;

```

TABLE II
THE HUB NODE SELECTION FUNCTION

- The JXTA network uses a universal resource binding mechanism called the resolver to perform all resolution operations found in traditional distributed systems, such as resolving a peer name into an IP address (DNS), binding a socket to a port, locating a service via a Directory service (LDAP), or searching for content in a distributed filesystem (NFS). In JXTA, all resolution operations are unified under the simple discovery of one or more advertisements

B. JXTA Pipes

Pipes are virtual communication channels used to send and receive messages between services and applications. Pipes provide a virtual abstraction over the peer endpoints to provide the illusion of virtual in and out mailboxes that are not physically bound to a specific peer location. A Propagate pipe connects one output pipe to multiple input pipes. Messages flow into the input pipe ends from the output pipe end (propagation source). The propagate message is sent to all listening input pipe ends in the current peer group context. Bi-directional, reliable and secure pipe services have been implemented on top of the core pipe services.

C. JXTA Security

The Project JXTA trust model, permits peers to be their own certificate authorities, or socially accumulated inter-peer interactions. Project JXTA provides strong cipher algorithms to protect principals such as local data (all local data is protected with a pass phrase), data in transit on the JXTA virtual network, and remotely stored data.

D. TRIUMF over JXTA

We have developed some of TRIUMF services on JXTA. These include Cryptographic SMC services, communication services and membership services. Based on these services, the secure aggregation framework mentioned

above has been implemented. The implementation details are given below. The JXTA implementation consists of following phases.

- **Phase 1:** Each node joins the default peer group (called as *netPeerGroup* in JXTA) to start with.
- **Phase 2:** If a node finds out that the number of his neighbors are more than a certain threshold, it tries to create a new sub-group.
- **Phase 3 (Sub Group creation):** Before a group gets created, the node makes sure that no sub-group has been created by any of its neighbors. If there is one, it joins that group. The creator of the group serves as Hub Node by default
- **Phase 4:** All the Hub nodes collaboratively compute a public private key pair (using threshold cryptosystem) such that private key is distributed among all hub nodes such that no two hub nodes know each others share of the private key. Each hub node publishes a Public Key service along with a BiDirectional communication Pipe (JXTABidiPipe) so that local peers can find the service and use the pipe to connect to the Hub node and get the public key.
- **Phase 5:** Each node in a group encrypts its value with that public key and broadcasts (using JXTAPropagatePipe) it to its immediate neighbors. In a gossip like protocol, all the encrypted values eventually reach the hub node of that group. This is an anytime algorithm meaning each node keeps on performing additively homomorphic encryption to its local and received values.
- **Phase 6:** All hub nodes themselves combine their accumulated encrypted values.
- **Phase 7:** Threshold decryption phase among hub nodes leave the instantaneous aggregate value over the entire network at the hub nodes.

V. CONCLUSIONS

We have made an attempt to develop a framework for trusted and secure collaborative computing. Practical realization is provided in the form of a middleware. Our focus is mainly on collaborations that require some form of sharing based on data processing. Full attempt is made to provide a comprehensive solution to the privacy problem in collaborative processes.

VI. FUTURE RESEARCH

Elliptic Curves are becoming increasingly important in many cryptographic situations. We are looking into Elliptic Curves based Efficient Homomorphic Cryptosystems. We are also developing different Secure Multiparty applications (Privacy Preserving Data Mining, Privacy Preserving Profile Matching etc) making use of TRIUMF services. Participants anonymity using Mix-nets, Dc-nets and Onion routing based concepts is being investigated.

REFERENCES

- [1] D. Vawdrey, T. Sundelin, K. E. Seamons, and C. Knutson. *Trust Negotiation for Authentication and Authorization in Healthcare Information Systems* 25th Annual International Conference of the IEEE Engineering In Medicine And Biology Society, Cancun, Mexico, September 2003.

- [2] William J. McEvily Jr., Aks Zaheer, Vincenzo Perrone, *Does Trust Matter? Exploring the Effects of Interorganizational and Interpersonal Trust on Performance*, Organization Science, 1998.
- [3] Gagan Aggarwal, Mayank Bawa, Prasanna Ganesan, Hector Garcia-Molina, Krishnaram Kenthapadi, Rajeev Motwani, Utkarsh Srivastava, Dilys Thomas, Ying Xu, *Two Can Keep A Secret: A Distributed Architecture for Secure Database Services*. Conference on Innovative Data Systems Research 2005: 186-199
- [4] Gagan Aggarwal, Toms Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, An Zhu, *Anonymizing Tables*. International Conference on Database Theory 2005: 246-258
- [5] Yehuda Lindell, Benny Pinkas: *Privacy Preserving Data Mining* J. Cryptology 15(3): 177-206 (2002)
- [6] Gagan Aggarwal, Nina Mishra, Benny Pinkas: *Secure Computation of the k-th-Ranked Element*. EUROCRYPT 2004: 40-55
- [7] Bobi Gilburd, Assaf Schuster, Ran Wolff: *Privacy-Preserving Data Mining on Data Grids in the Presence of Malicious Participants*. HPDC 2004: 225-234
- [8] Chris Clifton, Murat Kantarcioglu, AnHai Doan, Gunther Schadow, Jaideep Vaidya, Ahmed K. Elmagarmid, Dan Suciu: *Privacy-preserving data integration and sharing*. DMKD 2004: 19-26
- [9] Murat Kantarcioglu, Chris Clifton: *Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data*. IEEE Trans. Knowl. Data Eng. 16(9): 1026-1037 (2004)
- [10] A. Yao: *Protocols for secure computations*. In Proceedings of the twenty-third annual IEEE Symposium on Foundations of Computer Science, pages 160-164. IEEE Computer Society, 1982.
- [11] O. Goldreich, S. Micali, and A. Wigderson: *How to play ANY mental game*. In Proceedings of the nineteenth annual ACM conference on Theory of computing, pages 218-229. ACM Press, 1987.
- [12] P. Paillier: *Public-Key Cryptosystems based on Composite Degree Residue Classes*, Advances in Cryptology - EUROCRYPT 99, LNCS volume 1592, pp. 223-238. Springer Verlag, 1999.
- [13] I. Damgard, and M. Jurik: *A Generalisation, a Simplification and some Applications of Pailliers Probabilistic Public-Key System, Public Key Cryptography (PKC 2001)*, LNCS 1992, pp. 119-136. Springer Verlag, 2001.
- [14] L. Sweeney. *k-anonymity: a model for protecting privacy*. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 557-570.
- [15] P. Th. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, A.-M. Kermarrec. *Lightweight Probabilistic Broadcast*, p. 0443, The International Conference on Dependable Systems and Networks (DSN'01), 2001.
- [16] Hidayet Ozgur Sanli, Suat Ozdemir, and Hasan am, *SRDA: Secure Reference-Based Data Aggregation Protocol for Wireless Sensor Networks*, Proc. of IEEE VTC Fall 2004 Conference, Sept. 26-29, 2004, Los Angeles, CA, USA.
- [17] David Kempe, Alin Dobra, and J. E. Gehrke. *Computing Aggregate Information using Gossip*. In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2003). Cambridge, MA, October 2003.
- [18] David Wagner. *Resilient Aggregation in Sensor Networks*. 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04), October 25, 2004