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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

Generating Private Recommendations Efficiently Using GAE Datastore and Data Packing

Niranjan Kumar Nakkala[1], Ch.Ram Mohan[2], Dr.N.V.Rao[3]

PG Scholar, Computer Science and Engineering, CVR College of Engineering, Hyderabad, India[1]

Associate Professor, Computer Science and Engineering, CVR College of Engineering, Hyderabad, India[2]

Professor, Computer Science and Engineering, CVR College of Engineering, Hyderabad, India[3]

Abstract: People use social networks to get in touch with other people, and create and share content that includes personal information, images, and videos. The service providers have access to the content provided by their users and have the right to process collected data and distribute them to third parties. A very common service provided in social networks is to generate recommendations for finding new friends, groups, and events using collaborative filtering techniques. The data required or the collaborative filtering algorithm is collected from various resources including users' profiles and behaviors. Online shopping services increase the likelihood of a purchase by providing personalized suggestions to their customers. To find services and products suitable to a particular customer, the service provider processes collected user data like user preferences and click logs. In all of the above services and in many others, recommender systems based on collaborative filtering techniques that collect and process personal user data constitute an essential part of the service. On one hand, people benefit from online services. On the other hand, direct access to private data by the service provider has potential privacy risks for the users since the data can be processed for other purposes, transferred to third parties without user consent, or even stolen. Recent studies show that the privacy considerations in online services seem to be one of the most important factors that threaten the healthy growth of e-business. Therefore, it is important to protect the privacy of the users of online services for the benefit of both individuals and business. Recommender systems have become an important tool for personalization of online services. Generating recommendations in online services depends on privacy-sensitive data collected from the users. Traditional data protection mechanisms focus on access control and secure transmission, which provide security only against malicious third parties, but not the service provider. This creates a serious privacy risk for the users. This paper aims to protect the private data against the service provider while preserving the functionality of the system. We used GAE Datastore for the processing of private data to generate private recommendations by introducing semi-trusted third party and using data packing.

Keywords: GAE Datastore, Privacy, Recommender Systems, Data Packing, Collaborative Filtering, Private Recommendations

I. INTRODUCTION

In the last decade, we have experienced phenomenal progress in information and communication technologies. Cheaper, more powerful, less power consuming devices and high bandwidth communication lines enabled us to create a new virtual world in which people mimic activities from their daily lives without the limitations imposed by the physical world. As a result, online applications have become very popular for millions of people. Personalization is a common approach to further improve online services and attract more users. Instead of making general suggestions for the users of the system, the system can suggest personalized services targeting only a particular user based on his preferences. [19] Since the personalization of the services offers high profits to the service providers and poses interesting research challenges, research for generating recommendations, also known as collaborative filtering, attracts attention both from academia and industry. The techniques to generate recommendations for users strongly rely on information gathered from the user. This information can be provided by the user himself as in profiles or the service provider can observe users' actions, such as click logs. On one hand, more user information helps the system to improve the accuracy of the recommendations. On the other hand, the information on the users creates a severe privacy risk since there is no solid guarantee for the service provider not to misuse the users' data. It is often seen that whenever a user enters the system, the service provider claims the ownership of the information provided by the user and authorizes itself to distribute the data to third parties for its own benefits. The need for privacy protection for online services, particularly those using collaborative filtering techniques, triggered research efforts in the past years. Among many different approaches, two main directions, which are based on data perturbation and cryptography, have been investigated primarily in literature. Previous studies suggest hiding the personal data statistically, which has been proven to be an insecure approach. These studies present a recommender system that is built on distributed aggregation of user profiles, which suffers from the trade-off between privacy and accuracy. Related work mainly proposed a method using differential privacy, which has a similar trade-off between accuracy and privacy. They present an agent system where trusted software and secure environment are required. It means privacy-preserving collaborative forecasting and benchmarking to increase the reliability of local forecasts and data correlations using cryptographic techniques. This study also presented cryptographic protocols to generate recommendations, which suffer from a heavy computational and

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

communication overhead. Few more studies also propose protocols based on cryptographic techniques, which are computationally more efficient than their counterparts. However, in their proposals the users are actively involved in the computations, which makes the overall construction more vulnerable to time-outs and latencies in the users' connections. Moreover, the computations that a single user has to perform involve encryptions and decryptions in the order of thousands, which makes the system expensive to run for the users.

II. LITERATURE SURVEY

A. Cloud Computing

Cloud computing is defined as:

"Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLA"

- Vaquero, L., L. Rodero-Merino, et al. (2008)

B. IBM Research on Homomorphic Encryption

Homomophic Encryption is defined as:

"Homomorphic encryption is a very desirable goal. Suppose you want to add two numbers that are stored in an encrypted file. Traditionally the only way to do it was to decrypt the file, add the two numbers and then re-encrypt the file. Of course, to do the addition you had to have access to the entire contents. [1], [2] This also meant that other people could access it while it was stored as plain text. There are lots of situations where it would be good if the data could be stored in encrypted form, say in the cloud, and still operated on without having to decrypt it. This is the goal of fully homomorphic encryption".

- Craig Gentry. (2009)

C. Microsoft Research on Homomorphic Encryption

Since cloud is evolving, it necessitates new form of interactions and input/output technologies. Microsoft is working on two different projects "Inside the Cloud" and "Cloud Faster". In the first project, Microsoft is researching on Cloud Mouse, which is an interactive device for cloud computing, can be used as a secure key and users would be able interact with data in the cloud as if they were in the cloud. The latter project is collaboration between Bing and Windows Core Operating System Network team where the team is working on building new suite of protocols and architecture which reduces the latency inside data centres and thus increases the speed of applications in the cloud (Microsoft, 2010).

Greatest Concerns Surrounding Cloud Adoption at Your Company

Security	45%
Integration with existing systems	26%
Loss of control over data	26%
Availability concerns	25%
Performance issues	24%
IT governance issues	19%
Regulatory/compliance concerns	19%
Dissatisfaction with vendor offerings/pricing	12%
Ability to bring systems back in-house	11%
Lack of customization opportunities	11%
Measuring ROI	11%
Not sure	7%
Other	6%

*Respondents selected up to three criteria.
SOURCE: CIO Research

Table. 1. Concerns Surrounding Cloud Adoption

Is Cloud Security Strong Enough Yet?

Vendors have not adequately addressed security concerns around on-demand offerings.	59%

SOURCE: CIO Research

Table. 2. Is Cloud Secure?

D. Google Cloud Platform

1) Google App Engine:

Google App Engine is an online platform to host your web applications. It is what is popularly called a cloud Computing Platform. Google App Engine lets us run web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as our traffic and data storage needs grow. With App Engine, there are no servers to maintain: We just upload our application, and it's ready to serve our users. We can serve our app from our own domain name (such as http://www.example.com/) using Google Apps. Or, we can serve our app using a free name on the appspot.com domain. We can share our application with the world, or limit access to members of our organization.

Google App Engine supports apps written in several programming languages. With App Engine's Java runtime environment, we can build our app using standard Java technologies, including the JVM, Java servlets, and the Java programming language - or any other language using a JVM-based interpreter or compiler, such as JavaScript or Ruby. App Engine also features a Python runtime environment, which includes a fast Python interpreter and the Python standard library. App Engine also features a PHP runtime, with native support for Google Cloud SQL and Google Cloud Storage that works just like using a local mySQL instance and doing local file writes. Finally, App Engine provides a Go runtime environment that runs natively compiled Go code. These runtime environments are built to ensure that our application runs quickly, securely, and without interference from other apps on the system.

With App Engine, we only pay for what we use. There are no set-up costs and no recurring fees. The resources our application uses, such as storage and bandwidth, are measured by the gigabyte, and billed at competitive rates. We control the maximum amounts of resources our app can consume, so it always stays within our budget.

2) Data store:

Storing data in a scalable web application can be tricky. A user could be interacting with any of dozens of web servers at a given time, and the user's next request could go to a different web server than the previous request. All web servers need to be interacting with data that is also spread out across dozens of machines, possibly in different locations around the world. With Google App Engine, we don't have to worry about any of that. App Engine's infrastructure takes care of all of the distribution, replication, and load balancing of data behind a simple API—and we get a powerful query engine and transactions as well.

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

App Engine's data repository, the *High Replication Datastore (HRD),* uses the Paxos algorithm to replicate data across multiple datacenters. Data is written to the Datastore in objects known as *entities.* Each entity has a *key* that uniquely identifies it. An entity can optionally designate another entity as its *parent;* the first entity is a *child* of the parent entity. The entities in the Datastore thus form a hierarchically structured space similar to the directory structure of a file system. An entity's parent, parent's parent, and so on recursively, are its *ancestors;* its children, children's children, and so on, are its *descendants.*

An entity without a parent is a *root entity.* The Datastore is extremely resilient in the face of catastrophic failure, but its consistency guarantees may differ from what we are familiar with. Entities descended from a common ancestor are said to belong to the same *entity group;* the common ancestor's key is the group's *parent key,* which serves to identify the entire group. Queries over a single entity group, called *ancestor queries,* refer to the parent key instead of a specific entity's key. Entity groups are a unit of both consistency and transactionality: whereas queries over multiple entity groups may return stale, eventually consistent results, those limited to a single entity group always return up-to-date, strongly consistent results.

3) BigTable:

BigTable is a compressed, high performance, and proprietary data storage system built on Google File System, Chubby Lock Service, SSTable (log-structured storage like LevelDB) and a few other Google technologies. It is not distributed outside Google, although Google offers access to it as part of its Google App Engine. Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large size: petabytes of data across thousands of commodity servers. Many projects at Google store data in Bigtable, including web indexing, Google Earth, and Google Finance. These applications place very different demands on Bigtable, both in terms of data size (from URLs to web pages to satellite imagery) and latency requirements (from backend bulk processing to real-time data serving). Despite these varied demands, Bigtable has successfully provided a flexible, high-performance solution for all of these Google products.

E. Recommender System

Recommender System as defined as:

"There is an extensive class of Web applications that involve predicting userresponses to options. Such a facility is called a recommendation system."[6]

1) Collaborative filtering:

Collaborative Filtering (CF) is commonly used in the E-Commerce realm for producing recommendation for various products. CF is based on the assumption that people with similar tastes prefer the same items. In order to generate a recommendation, CF initially creates a neighborhood of users with the highest similarity to the user whose preferences are to be predicted. Then, it generates a prediction by calculating a normalized and weighted average of the ratings of the users in the neighborhood. [7], [17] In CF, user profile is a feature-vector containing information about user preferences with respect to a set of item the user rated. For quite some time CF has been applied in E-Commerce and direct recommendations of various kinds. Personalized information delivery in general and purchase recommendations (that applies collaborative filtering) in particular can increase the likelihood of a customer making a purchase, compared to non-personalized approaches. However, personalization brings with it the issue of privacy. Privacy is an important challenge facing the growth of Internet and the acceptance of various transaction models supported by Internet.

2) Privacy Enhanced Recommender System:

Recommender systems are widely used in online applications since they enable personalized service to the users. The underlying collaborative filtering techniques work on user's data which are mostly privacy sensitive and can be misused by the service provider. [19] To protect the privacy of the users, encrypt the privacy sensitive data and generate recommendations by processing them under encryption. With this approach, the service provider learns no information on any user's preferences or the recommendations made. The method is based on homomorphic encryption schemes and secure multiparty computation (MPC) techniques.

III. DRAWBACKS OF EXISTING STRATEGIES

Recommender systems based on collaborative filtering techniques that collect and process personal user data constitute an essential part of the service. On one hand, people benefit from online services. On the other hand, direct access to private data by the service provider has potential privacy risks for the users since the data can be processed for other purposes, transferred to third parties without user consent, or even stolen. Recent studies show that the privacy considerations in online services seem to be one of the most important factors that threaten the healthy growth of e-business. Therefore, it is important to protect the privacy of the users of online services for the benefit of both individuals and business. The need for privacy protection for online services, particularly those using collaborative filtering techniques, triggered research efforts in the past years.

Among many different approaches, two main directions, which are based on data perturbation and cryptography, have been investigated primarily in literature. Also propose protocols based on cryptographic techniques, which are computationally more efficient than their counterparts. However, in their proposals the users are actively involved in the computations, which makes the overall construction more vulnerable to time-outs and latencies in the users' connections. Moreover, the computations that a single user has to perform involve encryptions and decryptions in the order of thousands, which makes the system expensive to run for the users. Traditional data protection mechanisms focus on access control and secure transmission, which provide security only against malicious third parties, but not the service provider. This creates a serious privacy risk for the users.

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

IV. WORKING OF THE PROPOSED APPLICATION

The Proposed Application works on the following lines:

1. User must have valid credentials to access the web application.
2. Whenever user makes a purchase through the application, system records the transaction summary and restricts the service providers from viewing the data.
3. By using GAE Datastore, user private data is encrypted and placed into cloud technology which prevents service provider from viewing the data.

The application is designed intelligently enough to generate some private recommendations to the user based on his/her previous transactions.

V. ADVANTAGES OF PROPOSED SYSTEM

Traditional data protection mechanisms focus on access control and secure transmission, which provide security only against malicious third parties, but not the service provider. This creates a serious privacy risk for the users.
This Application is to protect the private data against the service provider while preserving the functionality of the system. We propose encrypting private data and processing them under encryption to generate recommendations. By introducing a semi-trusted third party and using data packing, we construct a highly efficient system that does not require the active participation of the user.

VI. SYSTEM ARCHITECTURE

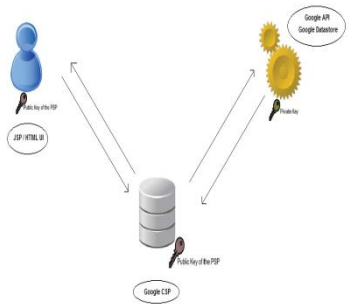

Fig. 1. System Architecture

VII. CLOUD SECURITY IMPLEMENTATION

The application is developed using Google App Engine SDK with JAVA language. Eclipse 3.6(java EE) IDE, Google Datastore is used for storing data. App Engine server is to handle client requests and the developed application is tested using different browsers such as FireFox, Internet Explorer, Google Chrome and Safari under Mac OS X and Windows platforms.

A. Admin Console

Login to 'Admin Console' to verify the entity present in the data store, Admin console can be opened on development machine using http://localhost:8888/_ah/admin Click on Data Store Viewer --> Select Item from Entity Kind list and click List Entities to view the product created.

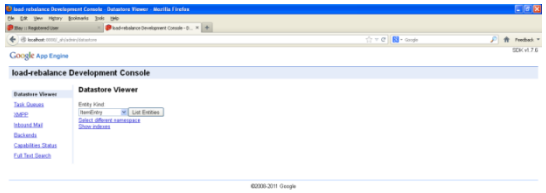

Fig. 2. Google App Engine Datastore Viewer

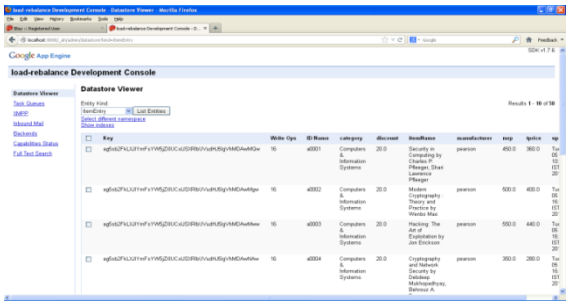

Fig. 3. Books Details in Google Datastrore

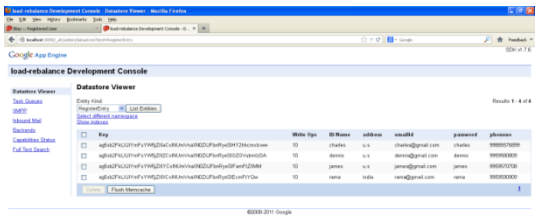

Fig. 4. Registered Users in Google Datastore

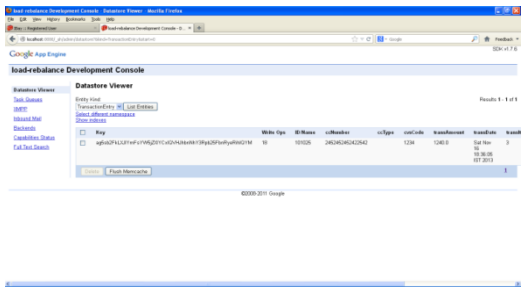

Fig. 5. Transaction Summary in Google Datastore (1)

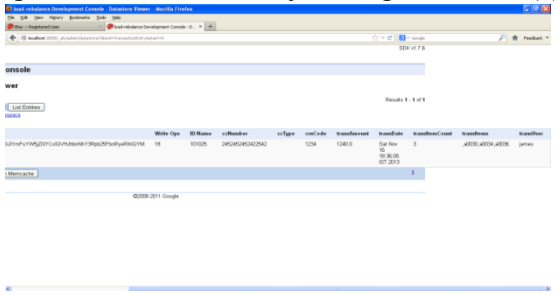

Fig. 6. Transaction Summary in Google Datastore (2)

The transaction summary of user is maintained separately in Google Datastore.

VIII. FINDINGS

The initial analysis on Homomorphic Encryption helped to understand various ways of exploiting it and become

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

aware of the current research works on security in Cloud Computing and Recommender System.

Recommender systems have become an important tool for personalization of online services. Generating recommendations in online services depends on privacy-sensitive data collected from the users. Traditional data protection mechanisms focus on access control and secure transmission, which provide security only against malicious third parties, but not the service provider. This creates a serious privacy risk for the users.

GAE Datastore is implemented for security in cloud computing which provides security to the users private data against the Google Cloud Service Provider, Google App Engine service simplified the implementation of GAE Datastore and generating private recommendations to users and we can protect the private data against the Service Provider.

IX. DISCUSSIONS

Direct access to private data by the service provider has potential privacy risks for the users since the data can be processed for other purposes, transferred to third parties without user consent, or even stolen. Recent studies show that the privacy considerations in online services seem to be one of the most important factors that threaten the healthy growth of e-business. Therefore, it is important to protect the privacy of the users of online services for the benefit of both individuals and business. Recommender systems have become an important tool for personalization of online services. Generating recommendations in online services depends on privacy-sensitive data collected from the users.

Traditional data protection mechanisms focus on access control and secure transmission, which provide security only against malicious third parties, but not the service provider. This creates a serious privacy risk for the users. This paper aims to protect the private data against the service provider while preserving the functionality of the system. We used GAE Datastore for the processing of private data to generate private recommendations by introducing semi-trusted third party and using data packing.

Our Application is to protect the private data against the service provider while preserving the functionality of the system. We proposed encrypting private data and processing them under encryption to generate recommendations. By introducing a semi-trusted third party and using data packing, we constructed a highly efficient system that does not require the active participation of the user.

X. LIMITATIONS

1. The system can run on any workstations running windows/linux.
2. User has to perform at least minimum no of transactions as defined for generating the recommendations and these recommendations cannot be viewed by the Google Cloud Service Provider.

We didn't consider the performance testing of this project as we expect the system would degrade in performance in case the user has huge volume of transaction history and it is out of scope of the project.

XI. CONCLUSION

Google Cloud Platform security is based on GAE Datastore, App Engine's data repository, the *High Replication Datastore (HRD)*, uses the Paxos algorithm to replicate data across multiple datacenters. Data is written to the Datastore in objects known as *entities*. Each entity has a *key* that uniquely identifies it. Where each user has unique security policy to retrieve and store data. Recommender systems are widely used in online applications since they enable personalized service to the users. The underlying collaborative filtering techniques work on user's data which are mostly privacy sensitive and can be misused by the service provider.

To protect the privacy of the users, encrypt the privacy sensitive data and generate recommendations by processing them under encryption. With this approach, the service provider learns no information on any user's preferences or the recommendations made. The method is based on homomorphic encryption scheme.

XII. SCOPE FOR FUTURE WORK

1. Practical implementation of Homomophic Encryption
2. Processing them under encrypted private data to generate private recommendations
3. Practical implementation of Homomorphic Encryption will provide complete security in the Cloud Computing
4. Generating more efficient recommendations by using Collaborative filtering
5. Hybrid Cloud Computing

ACKNOWLEDGEMENTS

I thank 'God', the almighty to have been with me throughout my life blessing me with energy and support towards the accomplishment of my goal. I thank my parents for their continuous motivation and love which acts as moral strength in achieving my goals. The research work could not have been this interesting and accomplished without the guidelines provided by my project supervisor Sri.Ch. Ram Mohan, Associate Professor, CSE. I would like to extend my gratefulness to him for his timely contribution of ideas and guidance which thereby paved way to achieve the work today.

Apart from the efforts and initiatives from me, the success of my work largely depends on the motivation and encouragement provided by all my faculty in the CVR College of Engineering. I also extend my thanks to all my friends who have been supportive achieving this work and special thanks to all my friends who reviewed this work. I also owe a big thanks to Centre for Development of Advanced Computing for the security conference held at Jawaharlal Nehru Technological University. This helped me to learn aspects of security oriented languages. I owe a

special thanks to Google Company, Oracle Corporation and Eclipse Foundation which made possible for me to gain access to recent version of Google App Engine SDK for Java, Java SE Development Kit 6 and Eclipse IDE for Java EE Developers.

The extensive support from my Professor Dr.N.V.Rao and college has to be mentioned. Support in the form of his guidelines and its exorbitant lab facilities, library, e-journal management and so on, without which it would have been intricate to precede the study; hence I would like to thank the **'**CVR College of Engineering**'** in whole for facilitating the students to a great extent.

REFERENCES

[1] Gentry, Craig. "A fully homomorphic encryption scheme." PhD diss.,Sta-nford University, 2009.
[2] Van Dijk, Marten, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan.
[3] "Fully homomorphic encryption over the integers." In Advances in Crypto-logy–EUROCRYPT 2010,pp. 24-43. Springer Berlin Heidelberg,2010.
[4] Gentry, Craig, and Shai Halevi. "Implementing gentry's fully-homomorphic encryption scheme." In Advances in Cryptology–EUROCRYPT 2011, pp. 129-148. Springer Berlin Heidelberg, 2011.
[5] Naehrig, Michael, Kristin Lauter, and Vinod Vaikuntanathan. "Can homo-morphic encryption be practical?." In Proceedings of the 3rd ACM works-hop on Cloud computing security workshop, pp. 113-124. ACM,2011.
[6] Erkin, Z. Dept. of Intell. Syst., Delft Univ. of Technol., Delft, Netherlands- Veugen, T. ; Toft, T. ; Lagendijk, R.L. "Generating Private Recommendations Efficiently Using Homomorphic Encryption and Data Packing" IEEE Transactions on Information Forensics and Security, vol.7, NO.3, June 2012
[7] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recom-mender systems: A survey of the state-of-the-art and possible extensions,"IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, Jun. 2005.
[8] N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis, Privacy risks in recommender systems," IEEE Internet Comput., vol. 5, no. 6, pp. 54–63, Nov./Dec. 2001.
[9] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in Proc.SI-GMOD Rec., May 2000, vol. 29, pp. 439–450.
[10] Y. Lindell and B. Pinkas, "Privacy preserving data mining," J. Cryptol.pp. 36–54, 2000, Springer-Verlag.
[11] H. Polat and W. Du, "Privacy-preserving collaborative filtering using ra-ndomized perturbation techniques.," in Proc. ICDM, 2003, pp. 625–628.
[12] H. Polat andW. Du, "SVD-based collaborative filtering with privacy," in Proc. 2005 ACM Symp. Applied Computing (SAC'05), New York, NY 2005, pp. 791–795, ACM Press.
[13] S. Zhang, J. Ford, and F. Makedon, "Deriving private information from randomly perturbed ratings," in Proc. Sixth SIAM Int. Conf. Data Mining, 2006, pp. 59–69.
[14] R. Shokri, P. Pedarsani, G. Theodorakopoulos, and J.-P. Hubaux, "Preserving privacy in collaborative filtering through distributed aggregation of offline profiles," in Proc. Third ACM Conf. Recommender Systems (RecSys'09), New York, NY, 2009, pp. 157–164, ACM.
[15] F.McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the net," in Proc. 15th ACM SIGKDD Int.Co-f. Knowledge Discovery and Data Mining (KDD'09), New York, NY,2-009, pp. 627–636, ACM.
[16] R. Cissée and S. Albayrak, "An agent-based approach for privacy preserving recommender systems," in Proc. 6th Int. Joint Conf. Autonomous A-gents and Multiagent Systems (AAMAS'07), New York, NY, 2007, pp.1–8, ACM.
[17] M.Atallah, M. Bykova, J. Li,K. Frikken, andM. Topkara, "Private collaborative forecasting and benchmarking," in Proc. 2004 ACM Workshop on Privacy in the Electronic Society (WPES'04), New York, NY, 2004, pp. 103–114, ACM.
[18] J. F. Canny, "Collaborative filtering with privacy.," in IEEE Symp. Security and Privacy, 2002, pp. 45–57.
[19] J. F. Canny, "Collaborative filtering with privacy via factor analysis," in
[20] SIGIR. New York, NY: ACM Press, 2002, pp. 238–245.
[21] [19] Z. Erkin, M. Beye, T. Veugen, andR. L. Lagendijk, "Privacy enhanced recommender system," in Proc. Thirty-First Symp. Information Theory in the Benelux, Rotterdam,2010, pp. 35–42.
[22] Z. Erkin, M. Beye, T. Veugen, and R. L. Lagendijk, "Efficiently computing private recommendations," in Proc. Int. Conf. Acoustic,Speech and Signal Processing (ICASSP), Prague, Czech Republic,May 2011, pp.5864- –5867, 2011.
[23] Abraham, D. (2009). "Why 2FA in the cloud?" Network Security 2009(-9): 4-5.
[24] Anonymous (2009). "Data in the cloud might be seized by government agencies without you knowing." Computer Fraud & Security 2009(8): 1.
[25] Bowers, K., A. Juels, et al. (2009). Hail: A high-availability and integrity layer for cloud storage, Proceedings of the 16th ACM conference on Computer and communications security 2009: 187-198.
[26] Cachin, C., I. Keidar, et al. (2009). "Trusting the Cloud." ACM SIGACT News 40(2).
[27] Chantry, D. (2009). "Mapping Applications to the Cloud." TechEd Special Edition 19: 2-9.
[28] Computing, D. and M. Creeger (2009). "Cloud Computing: An Overview ." Distributed Computing **7**(5).
[29] Cunsolo, V., S. Distefano, et al. (2009). "Cloud@Home: Bridging the Gap between Volunteer and Cloud Computing", LNCS 2009: 423-432.
[30] Cusumano, M. (2009). "An analysis of the cloud computing platform", Massachusetts Institute of Technology.
[31] Deelman, E., G. Singh, et al. (2008). "The cost of doing science on the cloud: the montage Example", Proceedings of the 2008 ACM/IEEE con-ference on Supercomputing 2008(50).
[32] Dikaiakos, M., D. Katsaros, et al. (2009). "Cloud Computing: Distributed Internet Computing for IT and Scientific Research." IEEE Internet Computing 3(5): 10-13.

BIOGRAPHIES

Niranjan Kumar Nakkala received B.Tech and M.Tech degrees in Co-mputer Science and Engineering from JNT University, Hyderabad, India. In Anwar-ul-uloom College of Engineering and Technology, 2010 and CVR College of Engineering (Autonomous Institution), 2013. His research interests include Information Security, Network Security and Cloud Computing.

Ch. Ram Mohan is currently working as Associate Professor in CSE Department at CVR College of Engineering, Hyderabad, India. He is having 13 years of experience in teaching. He obtained M.Tech (CS) from IETE, and He is pursuing Ph.D from JNT University, Hyderabad. His research interests include Ad-Hoc Networks and Information Security.

Dr N V Rao superannuated from DR-DO, after serving for more than 35 years. He obtained his M Tech (CSE) from Osmania University, and Ph.D from JNT University, Hyderabad. Currently working as a Professor, in CVR college of Engineering, Ibrahimpatnam, Hyderabad, India.