# A Survey On Blocking Hacking Users In Cloud Network Using Ticket Based Security Manager

Mr. Mohammed Riyaj
*4th Sem MTech (SE), CV Raman University*
*Kota Bilaspur(C.G.)*

Mr. Rohit Miri
ASSISTANT PROFESSOR
*DEPTT.OF ENGINEERING,(CSE)*
*Dr.C.V.RAMAN*
*UNIVERSITY,BILASPUR(C.G),INDIA*

Mr. Tarun Dhar Diwan
*ASSISTANT PROFESSOR*
*DEPTT.OF ENGINEERING (CSE)*
*Dr.C.V.RAMAN*
*UNIVERSITY,BILASPUR(C.G),*INDIA

*Abstract*— ***Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. However, the fact that user no longer have physical possession of the possibly large size of outsourced data makes the data integrity protection in Cloud Computing a very challenging and potentially formidable task, especially for users with constrained computing resources and capabilities. Thus, enabling public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met: 1) TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user; 2) The third party auditing process should bring in no new vulnerabilities towards user data privacy. We describe a privacy manager for cloud computing, which reduces the risk to the cloud computing user of their private data being stolen or misused, and also assists the cloud computing provider to conform to privacy law. We describe different possible architectures for privacy management in cloud computing; give an algebraic description of obfuscation, one of the features of the privacy manager; and describe how the privacy manager might be used to protect private metadata of database.***

***Keywords*— Cloud network, Bblacklisting, Third party auditor, Revocation, Ticket Method**

## I. INTRODUCTION

In this paper we describe a privacy manager for cloud computing, which reduces the risk to the cloud computing user of their private data being stolen or misused, and also assists the cloud computing provider to conform to privacy law. Cloud computing, in which services are carried out on behalf of customers on hardware that the customers do not own or manage, is an increasingly fashionable business model. The input data for cloud services is uploaded by the user to the cloud, which means that they typically result in users' data being present in unencrypted form on a machine that the user does not own or control. This poses some inherent privacy challenges. There is a risk of data theft from machines in the cloud, by rogue employees of cloud service providers or by data thieves breaking into service providers' machines, or even by other customers of the same service if there is inadequate separation of different customers' data in a machine that they share in the cloud. Governments in the countries where the data is processed or stored may have legal rights to view the data under some circumstances. There is also a risk that the data may be put to unauthorized uses. It is part of the standard business model of cloud computing that the service provider may gain revenue from authorized secondary uses of user's data, most commonly the targeting of advertisements. However, some secondary data uses would be very unwelcome to the data owner (such as, for example, the resale of detailed sales data to their competitors). At present there are no technological barriers to such secondary uses. There are, however, some legal constraints on the treatment of users' private data by cloud computing providers. Privacy laws vary according to jurisdiction, but EU countries generally only allow personally-identifiable information to be processed if the data subject is aware of the processing and its purpose, and place special restrictions on the processing of sensitive data (for example, health or financial data), the explicit consent of the data owner being part of a sufficient justification for such processing. They generally adhere to the concept of data minimization, that is, they require that personally identifiable information is not collected or processed unless that information is necessary to meet the stated purposes. In Europe, data subjects can refuse to allow their personally identifiable data to be used for marketing purposes. Moreover, there may be requirements on the security and geographical location of the machines on which personally identifiable data is stored. A UK business processing data about individual customers with some cloud computing services could find itself in breach of UK data processing law, if these services do not give assurances that the machines they use are adequately secure. European law limiting cross-border data transfers also might prohibit the use of the cloud computing services to process this data if they stored data in countries with weak privacy protection laws.

## II. RELATED WORK

Since in this paper we are interested in managing the privacy of data which is sent to a database in the cloud, in this section we place this work in a wider context by reviewing previous general approaches to privacy management for data repositories, for which various techniques have been developed to ensure that stored data is accessed in a privacy compliant way.

Some mechanisms and solutions have been built to encrypt confidential data when it is stored in data repositories, for example solutions using Translucent Databases. Most of these solutions focus on confidentiality and access control aspects, and have little flexibility in providing policy-driven mechanisms encompassing aspects beyond authentication and authorization. Describe access control policy-based encryption mechanisms for XML documents. Describes mechanisms for fine-grained encryption of parts of XML documents, in which decryption keys can either be granted to data receivers or collected from LDAP servers, based on data receivers' credentials. Focuses on related cryptographic mechanisms.

Hippocratic Databases include mechanisms for preserving the privacy of the data they manage. Their proposed architecture is based on the concept of associating privacy metadata (i.e. privacy policies) to data stored in data repositories, along with mechanisms to enforce privacy. The drawback of this approach is that it might require substantial changes to current data repository architectures, and therefore might take a long time and require substantial investment (by all the involved parties) to succeed. In addition, this approach does not take into account that the management of privacy spans across the database boundaries: such management has to be carried out within a broader context within cloud computing.

Although now withdrawn from production, IBM Tivoli Privacy Manager provided mechanisms for defining fine-grained privacy policies and associating them with data. The privacy policies contain authorization constraints along with constraints on contextual information and intent. This approach addressed the privacy management problem purely from an access control perspective within a single enterprise. It did not include additional aspects relevant for privacy management within cloud computing such as trust management and dealing with ongoing privacy obligations dictated by legislation and enterprises' guidelines.

An alternative approach is based on an adaptive privacy management system where data are retrieved from standard data repositories, and parts of these data are encrypted and associated with privacy policies. This aims to make use of current data repository technologies and reduce to the minimum the impact on them, in terms of required changes: interactions with data repositories can still happen but in a way that confidential data is protected and contextually released, in a fine-grained way, based on the fulfilment of associated privacy policies.

## III. Problem Statement

We consider a cloud data storage service involving three different entities, as illustrated in Fig.1: the cloud user (U), who has large amount of data files to be stored in the cloud; the cloudserver (CS), which is managed by cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter.); the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. We consider the existence of a semi-trusted CS in the sense that in most of time it behaves properly and does not deviate from the prescribed protocol execution. While providing the cloud data storage based services, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. We assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users. However, any possible leakage of user's outsourced data towards TPA through the auditing protocol should be prohibited. Note that to achieve the audit delegation and authorize CS to respond to TPA's audits, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate. These authentication handshakes are omitted in the following presentation.

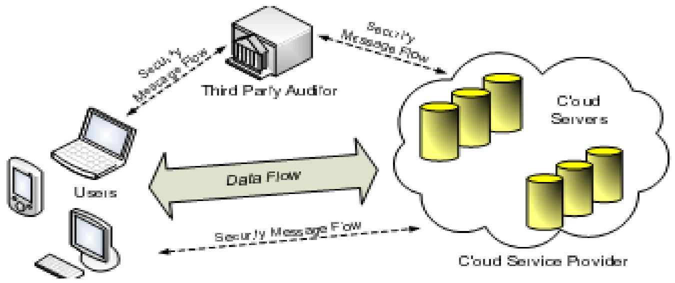

*Fig 1: Architecture of Cloud Data Storage Service*

## IV. Proposed System

We present a secure system called Security Manager (SM), which provides all the following properties: anonymous authentication, backward unlinkability, subjective blacklisting, fast authentication speeds, rate-limited anonymous connections, revocation auditability (where users can verify whether they have been blacklisted), and also addresses the Sybil attack to make its deployment practical. In SM, users acquire an ordered collection of tickets, a special type of pseudonym, to connect to websites. Without additional information, these tickets are computationally hard to link, and hence using the stream of tickets simulates anonymous access to services.

Websites, however, can blacklist users by obtaining a seed for a particular Ticket, allowing them to link future ticket from the same user — those used before the complaints remain unlikable. Servers can therefore blacklist anonymous users without knowledge of their IP addresses while allowing behaving users to connect anonymously. Our system ensures that users are aware of their blacklist status before they present a Ticket, and disconnect immediately if they are blacklisted. Although our work applies to cloud networks in general, we consider Tor for purposes of exposition. In fact, any number of cloud networks can rely on the same Security system, blacklisting anonymous users regardless of their cloud network(s) of choice.

An anonymous P2P communication system is a peer-to-peer distributed application in which the nodes or participants are anonymous or pseudonymous .Anonymity of participants is usually achieved by special routing overlay networks that hide the physical location of each node from other participants. Interest in anonymous P2P systems has increased in recent years for many reasons, ranging from the desire to share files without revealing one's network identity and risking litigation to distrust in governments, concerns over mass surveillance and data retention, and lawsuits against bloggers.

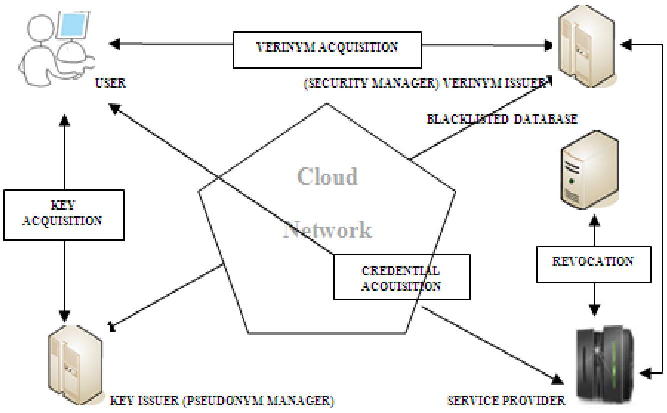

*Fig 2- The Proposed system architecture showing the various modes of interaction.*

### A. *Pseudonym & Security Manager*

Direct contact of the user is mandatory towards the pseudonym manager for demonstrating control over a resource. Same pseudonyms are constantly issued for the same resource. The pseudonym manager's assignments are constrained to mapping IP addresses to pseudonyms. The user contacts the pseudonym manager only once per likability window.

The process starts with the connection to the security manager, after obtaining a pseudonym by the user via cloud network. The user's requests to the security manager are pseudonyms and tickets are specific to a particular user-server pair. The system cannot identify the specific user and the connected server. Until the pseudonym and security manager do not collude. That shows the security manager is familiar with only the pseudonym-server pair and the pseudonym manager deals only with the user identity-pseudonym pair.

### B. *Blacklisting a User & Blacklisting Status*

In case a user misbehaves; any future connection may be linked by the server within the current linkability window. The provision of backward linkability and subjective blacklisting are facilitated, because the user's past connections remain unlinkable inspite of the future blocking of the misbehaving user.

In the present system, the facility of notification of the blacklist status is possible, by downloading the server's blacklist; a user can verify the status and immediately disconnect it. The authenticity of the blacklist can easily be verified, provided that the list is updated in the current time period. If it is not updated as such, the "daisies" provided by security manager ensures the updated version.

We can be sure about the non existence of race conditions in the verification of freshness of a blacklist, due to the use of 'digital signatures' and 'daisies'. In the updates to the ticket protocol the privacy properties associated with ticket alone had already been proved as part of a two-tiered hash chain. Now the security at the protocol level is to be proved. It is a process of redesigning and refining the definitions of the protocols to protect against towards privacy.

As such a large anonymity sets are created by preventing the server from distinguishing between the users already connected in the same time period and those who are blacklisted. By this process, servers obtain proofs of freshness every time period for easy download verification.
To assure efficiency of the blacklist updating, lightweight daisies are issued by SM to servers as proof of freshness. The SM embeds a distinct identifier ticket for direct recognition. Time is divided into linkability windows of duration W, each of which is split into L time periods of duration T (i.e., W=L*T)

## V. Design Goals

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantees.

1) Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users.

2) Storage correctness: to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact.

3) Privacy-preserving: to ensure that the TPA cannot derive users' data content from the information collected during the auditing process.

4) Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.

5) Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

## VI. Conclusion & Future Work

In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing, where TPA can perform the storage auditing without demanding the local copy of data. We utilize the homomorphism authenticator and random mask technique to guarantee that TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where TPA can perform the multiple auditing tasks in a batch manner, i.e., simultaneously. Extensive security and performance analysis shows that the proposed schemes are provably secure and highly efficient. We believe all these advantages of the proposed schemes will shed light on economies of scale for Cloud Computing.

## References

[1] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren, Wenjing Lou, "Privacy Preserving Public Auditing for Secure Cloud Storage", IEEE 2013

[2] Mohamed Nabeel, Ning Shang, Elisa Bertion, "Privacy Preserving Policy Based Content Sharing In Public Cloud", IEEE 2012

[3] Kui Ren, Cong Wang, Qian Wang, "Security Challenges for Public Cloud", IEEE 2010

[4] Hassan Takabi, James B. D. Joshi, Gail, "Security and Privacy Challenges in Cloud Computing Environment", IEEE 2010

[5] Cong Wang, Qian Wang, Kui Ren, Wenjing Lou, "Privacy Preserving Public Auditing Data Storage Security in Cloud Computing", IEEE Proceedings 2010

[6] Patrick P.Tsang, Apu Kapadia, Cory Cornelius, Sean W.Smith, "Nymble: Blocking Misbehaving Users in Anonymizing Networks", IEEE Transaction on Dependable and Secure Computing, Vol-8, No.2, March-April 2011

[7] Reed S. Abbott, TimothyW. van der Horst, and Kent E. Seamons. CPG: Closed Pseudony-mous Groups. In Vijay Atluri and Marianne Winslett, editors, Proceedings of WPES 2008, pages 55–64. Association for Computing Machinery (ACM) Press, New York, NY, USA, October 2008. (One citation on page 17.)

[8] Peter C. Johnson, Apu Kapadia, Patrick P. Tsang, and Sean W. Smith. Nymble:Anonymous IP-Address Blocking. In Privacy Enhancing Technologies, LNCS 4776, pages 113–133. Springer, 2007.

[9] P.P. Tsang, M.H. Au, A. Kapadia, and S.W. Smit "Blacklistable Anonymous Credentials: Blockin Misbehaving Users without TTPs," Proc. 14th ACM Con Computer and Comm.Security (CCS '07), pp. 72-81, 2007.

[10] Tadayoshi Kohno, Andre Broido, and K. C. Claffy. Remote physical device finger-printing. In Proceedings of the 2005 IEEE Symposium on Security and Privacy, pages 211–225, Washington, DC, USA, 2005. IEEE Computer Society.

[11] Toru Nakanishi and Nobuo Funabiki. Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps. In ASIACRYPT, LNCS3788, pages 533–548. Springer, 2005.

[12] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of Group Signatures: TheCase of Dynamic Groups. In CT-RSA, LNCS 3376, pages 136–153. Springer, 2005.

[13] Dan Boneh and Hovav Shacham. Group Signatures with Verifier-Local Revocation. In ACM Conference on Computer and Communications Security, pages 168–177. ACM, 2004.

[14] RogerDingledine, Nick Mathewson, Paul Syverson,"Tor: The Second-Generation Onion Router," Proc.Usenix Security Symposium, pp. 303-320, 2004.

[15] A. Kiayias, Y. Tsiounis, and M. Yung, "Traceable Signatures," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT), Springer, pp. 571-589, 2004.

[16] Jan Camenisch and Anna Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In CRYPTO, LNCS 3152, pages 56–72. Springer, 2004.

[17] I. Teranishi, J. Furukawa, and K. Sako, "k-Times Anonymous Authentication (Extended Abstract)," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), Springer, pp. 308-322, 2004.

[18] Giuseppe Ateniese, Dawn Xiaodong Song, and Gene Tsudik. Quasi-Efficient Revo-cation in Group Signatures. In Financial Cryptography, LNCS 2357, pages 183–197.Springer, 2002.

[19] Jan Camenisch and Anna Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In EUROCRYPT,LNCS 2045, pages 93–118. Springer, 2001.

[20] Emmanuel Bresson and Jacques Stern. Efficient Revocation in Group Signatures. In Public Key Cryptography, LNCS 1992, pages 190–206. Springer, 2001.