Prevention of Insider Attacks by Integrating Behavior Analysis with Modified Risk based Access Control Model to Protect Cloud

Simranjeet Kaur
Department of Computer Science and Engineering
Sri Sai College of Engineering and Technology, Badhani Pathankot, Punjab, India

Prof. Meenakshi Sharma
Head of Computer Science and Engineering
Sri Sai College of Engineering and Technology, Badhani Pathankot, Punjab, India

Sukhbeer Singh
Department of Computer Science and Engineering
Sri Sai College of Engineering and Technology, Badhani Pathankot, Punjab, India

Abstract - Cloud is being used commonly now days by number of users. The intensions of the users will be indifferent. The cloud security will be considered by the developer so that the data stored within the cloud cannot be at stakes. Cloud security will be enhanced by the use of proposed techniques. The technique will help in establishing the secure way by which data is accessed by the users. The cloud security will be accomplished by the use of keystroke analyzer. This technique will analyze the pattern through which user is pressing the keys within the existing system. If the pattern is different from the existing pattern than system will cause the security alerts.

Keywords – Cloud, Keystroke Analyzer, Pattern.

I. INTRODUCTION

One of the important challenges in MCC [21, 22] is security and privacy [41]. Furthermore, authentication plays an important role in preserving security and privacy of mobile communication especially in wide spread networks such as MCC [37, 38, 40]. It helps to protect shared information from unauthorized persons. In other words, an authentication mechanism determines how user identified and verified to access to sensitive information [42]. Verification of user's identity is the most important goal behind an authentication. PIN is adopted as the only security mechanism for mobile devices. It is obvious that, PIN (something the user knows) is not very secure mechanism for authenticating users because of its limitation, as well as it is difficult to confirm that the demand is from the rightful owner [17, 44]. Strong method of authentication should cover one or several various factors of identification to improve security. These factors are i) something we know; ii) something we have; iii) something we are. Therefore, biometric authentication [11, 13] is a strong authentication mechanism by providing the factor what we are and what we know [28]. In addition, it is able to identify users based on their unique characteristic [35], and it is more reliable, because it is so difficult for user to pretend as other user by using physical or behavioral biometric authentication. Keystroke authentication is a type of behavioral biometric authentication. Keystroke based authentication can categorize in two folds, Keystroke Static Authentication (KSA), as well as KDA. Keystroke static authentication can identify keystroke of users only at particular times, for example the time that user wants to login. This is a huge drawback of KSA; due to system can use by anyone once the user is authenticated at login [36]. Majority of the researches have focused on KSA using inter-keystroke latency mechanism [9]. Static authentication provides more strong and robust user authentication than simple password or PIN; however it cannot keep continuous security. KDA continuously observes the style of typing of the users throughout the whole stage of interaction even after a successful login. In other words, the typing patterns of users are constantly analyzed and when they do not match accessing of users will block [9, 34]. The main goal of KDA is recognizing mobile users by identifying and analyzing their unique feature for authentication such as typing pressure, keystroke duration, typing error, and latency of keystrokes [9]. KDA has some advantages rather than other types of biometric authentication. These advantages are i) Contrasting other biometric methods, KDA does not need any additional tools, therefore it causes to decrease price, ii) High acceptability between mobile users due to it is natural for everybody to type a password for authentication purposes, iii) preserving privacy and security of users because it is based on behavioral characteristic of users, iv) It could not be forgotten, stolen or lost [28, 32]. This paper is organized as follows:

Section 2 discusses the various researches related to biometric authentication, as well as keystroke base authentication. Explanation about the proposed method is in Section 3. Experimental results obtained from applying KDA in CSP using Android application development bring in Section 4; finally the conclusions are given in Section 5.

II. RELATED WORK

Related Work Recently, much of growing interest has been pursued in the context of remotely stored data verification [1–9, 11, 13–15]. Ateniese et al. [2] define the "provable data possession" (PDP) model for ensuring possession of files on untrusted storages. In their scheme, they utilize RSA-based homomorphic tags for auditing outsourced data, thus can provide public verifiability. However, Ateniese et al. do not consider the case of dynamic data storage, and the direct extension of their scheme from static data storage to dynamic case brings many design and security problems. In their subsequent work [11], Ateniese et al. propose a dynamic version of the prior PDP scheme. However, the system imposes a priori bound on the number of queries and does not support fully dynamic data operations, i.e., it only allows very basic block operations with limited functionality and block insertions cannot be supported. In [13], Wang et al. consider dynamic data storage in distributed scenario, and the proposed challenge-response protocol can both determine the data correctness and locate possible errors. Similar to [11], they only consider partial support for dynamic data operation. Juels et al. [3] describe a "proof of retrievability" (PoR) model and give a more rigorous proof of their scheme. In this model, spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on archive service systems. Specifically, some special blocks called "sentinels" are randomly embedded into the data file F for detection purpose and F is further encrypted to protect the positions of these special blocks. However, like [11], the number of queries a client can perform is also a fixed priori and the introduction of pre-computed "sentinels" prevents the development of realizing dynamic data updates. In addition, public verifiability is not supported in their scheme. Shacham et al. [1] design an improved PoR scheme with full proofs of security in the security model defined in [3]. Like the construction in [2], they use publicly verifiable homomorphic authenticators built from BLS signatures [16] and provably secure in the random oracle model. Based on the BLS construction, public retrievability is achieved and the proofs can be aggregated into a small authenticator value. Still the authors only consider static data files. Erway et al. [14] was the first to explore constructions for dynamic provable data possession. They extend the PDP model in [2] to support provable updates to stored data files using rank-based authenticated skip lists. This scheme is essentially a fully dynamic version of the PDP solution. In particular, to support updates, especially for block insertion, they try to eliminate the index information in the "tag" computation in Ateniese's PDP model [2]. To achieve this, before the verification procedure, they employ authenticated skip list data structure to authenticate the tag information of challenged or updated blocks first. However, the efficiency of their scheme remains in question. It can be seen that while existing schemes are proposed to aiming at providing integrity verification under different data storage systems, the problem of supporting both public verifiability and data dynamics has not been fully addressed. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in cloud computing.

III. PROPOSED SYSTEM

The proposed system will consist of keystroke analyzer which could test the frequency of each character which is used within the cloud by the user. The pattern analyzer will also be used in this case. The pattern analyzer will go to determine weather pattern is malicious or not. The malicious pattern if detected will be rejected. The concept of deduplication is also used within the proposed system so that total bandwidth consumption can also be reduced. pose risk of cloud storage services on behalf of the clients upon request. In the cloud paradigm, by putting the large data files on the remote servers, the clients can be relieved of the burden of storage and computation. As clients no longer possess their data locally, it is of critical importance for the clients to ensure that their data are being correctly stored and maintained. That is, 5 clients should be equipped with certain security means so that they can periodically verify the correctness of the remote data even without the existence of local copies. In case that clients do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the monitoring task to a trusted TPA. In this paper, we only consider verification schemes with public verifiability: any TPA in possession of the public key can act as a verifier. We assume that TPA is unbiased while the server is untrusted. Note that we don't address the issue of data privacy in this paper, as the topic of data privacy in Cloud Computing is orthogonal to the problem we study here. For application purposes, the clients may interact with the cloud servers via CSP to access or retrieve their pre-stored data. More importantly, in practical scenarios the client may frequently perform block-level operations on the data files. The most general forms of these operations we consider in this paper are modification, insertion, and deletion.
The proposed system has following output associated with it.

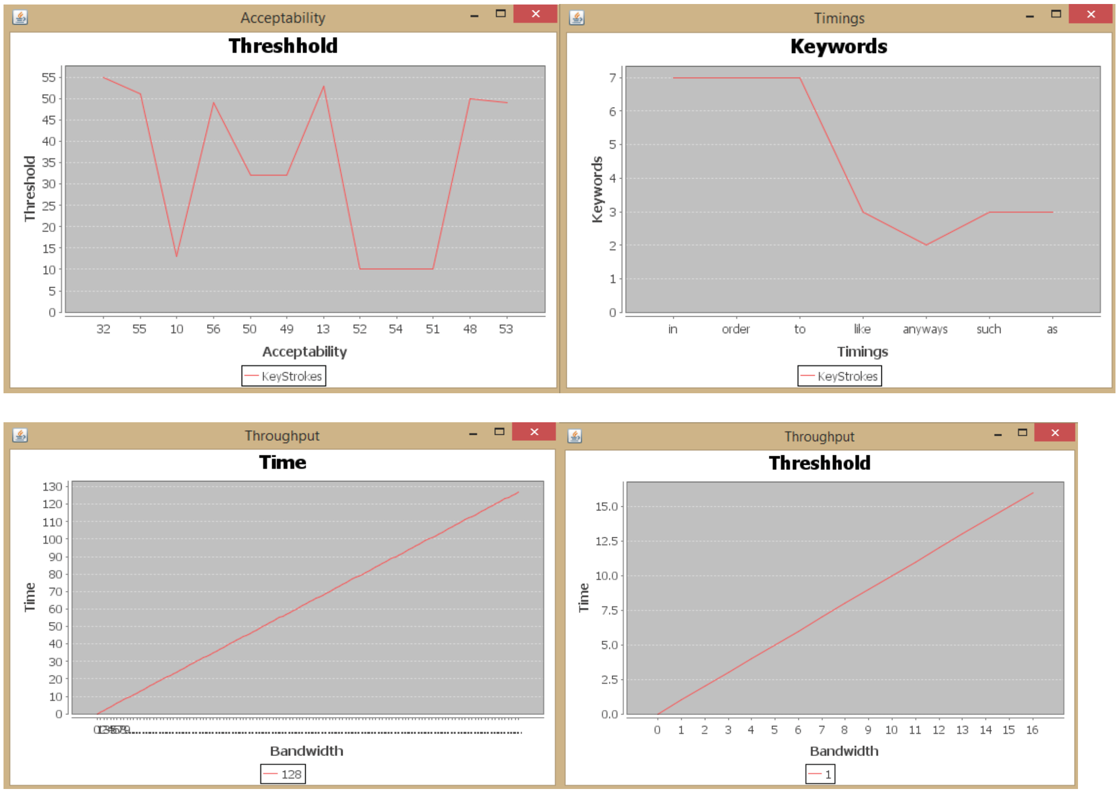

Figure 1. Showing the output of the Keystroke analyzer.

IV. CONCLUSION AND FUTURE WORK

To ensure cloud data storage security, it is critical to enable a third party auditor (TPA) to evaluate the service quality from an objective and independent perspective. Public verifiability also allows clients to delegate the integrity verification tasks to TPA while they themselves can be unreliable or not be able to commit necessary computation resources performing continuous verifications. Another major concern is how to construct verification protocols that can encounter cost. Comparison of communication complexity between our RSA-based instantiation and DPDP [14], for 1 GB file with variable block sizes. The detection probability is maintained to be 99%. modate dynamic data files. In this paper, we explored the problem of providing simultaneous public verifiability and data dynamics for remote data integrity check in Cloud Computing. Our construction is deliberately designed to meet these two important goals while efficiency being kept closely in mind. We extended the PoR model [1] by using an elegant Merkle hash tree construction to achieve fully dynamic data operation. Experiments show that our construction is efficient in supporting data dynamics with provable verification.

REFERENCES

[1] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of ASIACRYPT'08. Springer-Verlag, 2008, pp. 90–107.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of CCS'07. New York, NY, USA: ACM, 2007, pp. 598–609.

[3] A. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in Proc. of CCS'07. New York, NY, USA: ACM, 2007, pp. 584–597. 20

[4] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," Cryptology ePrint Archive, Report 2008/175, 2008.

[5] M. Naor and G. N. Rothblum, "The complexity of online memory checking," in Proc. of FOCS'05, 2005, pp. 573–584.

[6] E.-C. Chang and J. Xu, "Remote integrity check with dishonest storage server," in Proc. of ESORICS'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 223–237.

[7] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[8] A. Oprea, M. K. Reiter, and K. Yang, "Space-efficient block storage integrity," in Proc. of NDSS'05, 2005.

[9] T. Schwarz and E. L. Miller, "Store, forget, and check: Using algebraic signatures to check remotely administered storage," in Proc. of ICDCS'06, 2006.

[10] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and secure sensor data storage with dynamic integrity assurance," in Proc. of IEEE INFOCOM'09, Rio de Janeiro, Brazil, Appril 2009.

[11] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. of SecureComm'08, 2008.

[12] C. Wang, K. Ren, and W. Lou, "Towards secure cloud data storage," Proc. of IEEE GLOBECOM'09, submitted on March 2009.

[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in Proc. of IWQoS'09, Charleston, South Carolina, USA, 2009.

[14] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," Cryptology ePrint Archive, Report 2008/432, 2008.

[15] K. D. Bowers, A. Juels, and A. Oprea, "Hail: A high-availability and integrity layer for cloud storage," Cryptology ePrint Archive, Report 2008/489, 2008.

[16] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in Proc. of ASIACRYPT'01. London, UK: Springer-Verlag, 2001, pp. 514–532.

[17] R. C. Merkle, "Protocols for public key cryptosystems," Proc. of IEEE Symposium on Security and Privacy'80, pp. 122–133, 1980.

[18] D. Boneh and C. Gentry, "Aggregate and verifiably encrypted signatures from bilinear maps," in Proc. of Eurocrypt'03, volume 2656 of LNCS. Springer-Verlag, 2003, pp. 416–432.