Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 7, July 2013, pg.327 – 332

RESEARCH ARTICLE

TPM as a Middleware for Enterprise Data Security

Abhishek Patel[1], Prabhat Dansena[2]
[1]M. Tech, CSE, India
[2]School of Computer and Information Sciences, UoH, India

[1] *mrabhi.patel@gmail.com*; [2] *p.dansena23@gmail.com*

Abstract— Cloud Computing is one of the emerging technologies in Computer Science. Cloud provides various types of services to us. In the Private Cloud Computing the major concern is to securing data/files and also providing privacy. Storage as a Service is generally seen as a good alternative for a small or mid-sized business that lacks the capital budget and/or technical personnel to implement and maintain their own storage. Currently, the service providers that provides cloud storage, like Dropbox provides security by server-side data encryption. Since all the encryption keys are managed by software, so such method is not secure enough. But the main issue is to maintain CIA (Confidentiality, Integrity and Authentication) to the data stored in the cloud. To achieve these we use Public Key Cryptography. For securing data we use Asymmetric key Encryption Technique. Key rings are provided by the TPM. We describe an architecture which protects enterprise data in cloud and also having authentication based on the signature. We use asymmetric keys for encrypting data. We will use the keys generated by Trusted Platform Module (TPM) for providing better security. Use of TPM is a more secure way to encrypt and decrypt data. So we have implemented a TPM as a middleware which applies the specification of Trusted Computing Group (TCG). TCG is a global industry standard, supportive of a hardware-based root of trust, for interoperable trusted computing platforms. We use TPM to encrypt data before storing it to the cloud. And we use have to use any basic Authentication Service to avoid masquerading, replay attack and eavesdropping to the client side.

Key Terms: - Security; TPM; TCG; Network; Cloud Storage

I. INTRODUCTION

Encryption is one of the major reasons which encouraged keeping online backup of crucial data. Encryption helps to avoids unauthorized access and attempts to corrupt files [2]. For this reason, hardware based Trusted Computing ceremoniously adopted by industries to protect their computing infrastructure.

TCG [1] created the Trusted Platform Module with cryptographic capabilities. With growth and development in computing devices, TCG has also extended its concept of trusted systems. It has moved well ahead of TPM-enabled computers, to all new range of devices like hard disks and mobile phones with TCG. Trusted Computing technologies developed by TCG are now deployed as a part of many enterprise systems, storage systems, networks, embedded systems, and mobile devices and can secure cloud computing and virtualized systems.

TPM (Trusted Platform Module) is a microcontroller based device that can provide security to the parameters that are used to authenticate the underlying platform and to validate the system. These parameters can range from passwords, certificates, or encryption keys. A TPM can store platform measurements that certify the platform. Authentication and Attestation are necessary steps for ensuring the safety and trust for the target

platform. Attestation provides guarantee the trust of platform has not been breached. TPM maintains the three (CIA) properties (i.e. Confidentiality, Integrity and Authentication).

II. METHODOLOGY

For the vast majority of cloud storage, the security and privacy options provided are perfectly acceptable. The fact is that most people just don't care about privacy. For those of us that do, however, there is a relatively easy solution that can allow you to continue using cloud storage and keep your important data secure, Using the Trusted Architecture, you can create encrypted folders within your cloud storage [11], which gets synchronized like any other file from the Trusted Architecture.

We usually have our data stored in multiple clouds like- Salesforce, Box.net, Gmail, Amazon etc. Trusted Architecture provides us the ability to create a uniform data protection scheme across all clouds. Similar to in-line security gateway that stands between users and cloud applications The Architecture is used to encrypt any data that flows out from the network. Application of encryption on a cloud security gateway, the architecture eliminates security and privacy risks of cloud computing.

Mostly data stored in cloud are not in protected format. There is a big concern of security in cloud storage. The Trusted Architecture [4] provides a way to encrypt sensitive information to the enterprises as it moves to any cloud application and then decrypt it again as data is delivered to end users. This protects the data from being accessed by others. This revolutionary technology maintains the cloud application user experience, with near zero latency, and without making any changes to the cloud application itself.

The architecture we have proposed is very advantageous:

- We are providing security to the end user using TPM. However End user has no TPM device installed in its system. It means end user doesn't require TPM in its machine and take advantages of TPM.
- The architecture we provide is easy to use. The end user has nothing to install or uninstall in its system for taking the benefit of the architecture.
- A single TPM device is used for multiple users. Hence we are taking extra benefit from TPM device.

III. DESIGN DETAILS

Fig. 1 and Fig. 2 shown below are showing the working flow of our design. In Fig. 1 we have a trusted gateway installed in between end user and cloud/remote storage. Every time when a user will request to store its data into cloud storage first it will be passed through trusted gateway. Then trusted gateway will store that data into cloud storage. Here trusted gateway is first encrypting the data of user using the legacy key created by TPM. After that trusted gateway is uploading the encrypted data into cloud storage.

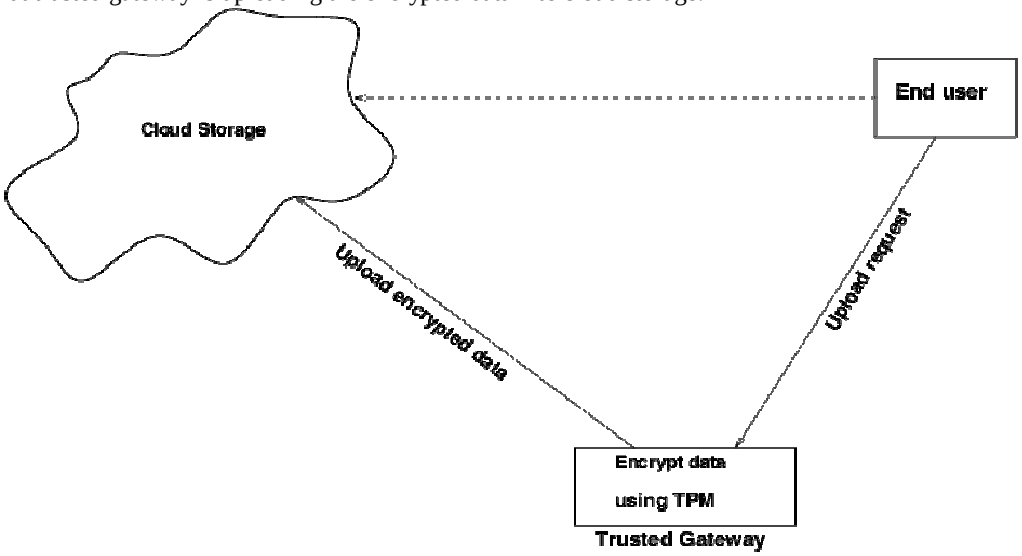

Fig. 1 Flow diagram for encryption of TPM based Trusted Architecture

In Fig. 2 it is shown that whenever a user will request to download his/her data, firstly the request will be redirected towards the trusted gateway and the cloud storage server will send the encrypted data to trusted gateway. Then after decrypting the user data it will be send to the user as a response from trusted gateway. In short the cloud storage is only keeping the encrypted data. Only trusted gateway can decrypt.

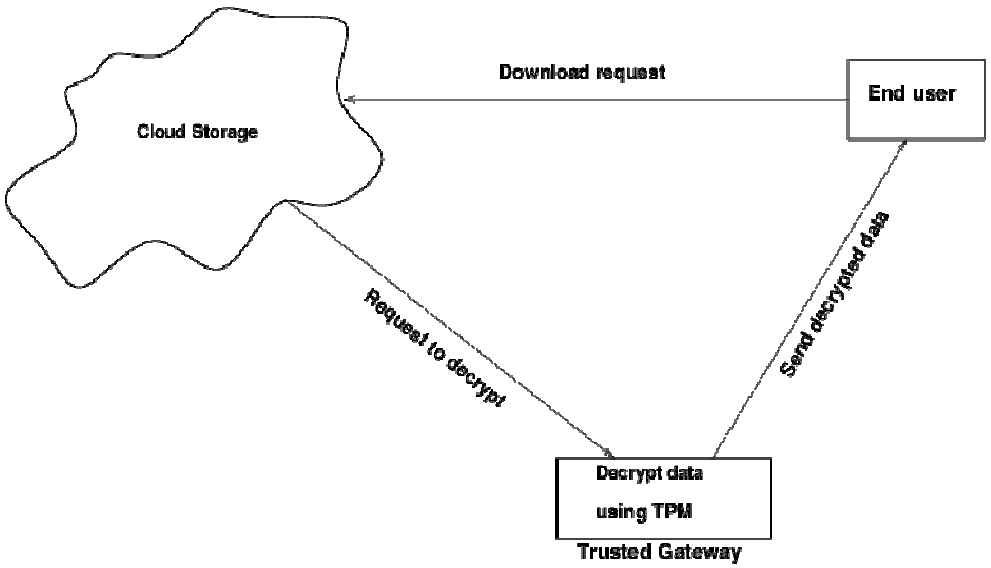

Fig. 2 Flow diagram for decryption of TPM based Trusted Architecture

IV. IMPLEMENTATION

In our implementation of the design shown above we have used two application servers, one is glassfish server as a TPM server working as a trusted gateway and an apache server as a cloud server working as a cloud storage. Using TPM server a user can upload its file to cloud server. We have used java library provided by MIT [3] for utilizing the TPM.

Following are the steps we have used to set up a TPM server.

- First we will create an Attestation Identity Key for platform authentication.
- Then we will create a legacy key. It will be used for unbinding the data.
- Loading the key created in the previous step into TPM and TPM will return its handle. Handle is basically a pointer to access the private portion of the key.

Inside TPM server we have also created an application using java which will provide an interface for uploading the data file for encryption and decryption.

In cloud server a file listing web application is installed whenever user will upload its data to TPM server. The TPM server will encrypt the user data and store that data into cloud server. User can view and download its data using the web application installed in the cloud server. When user will download his/her data firstly the data will pass through the TPM server. After that TPM server will unbind the user's encrypted data using the key loaded into the TPM. Then finally return that decrypted data file to the user.

V. RESULTS AND DISCUSSIONS

The proposed architectural scheme is aimed at providing TPM based data security to the users of the network. This is intended to help the clients to store their data on cloud; in encrypted form. All the requests/data are forwarded to a TPM server which is equipped with TPM chip to encrypt (or decrypt) the data when storing (retrieving) data from the cloud store.

The basic requirement of such a scheme requires us to study the timing of various events in the process of encryption (decryption). In all the presented results network delays are not considered as they are the part of network latency and not the part of encryption/decryption system.

As discussed in the methodology part, the TPM server might be servicing multiple requests at any given time. The arrival rate of these processes cannot be determined as it depends on the number of clients and network latency. But we can surely define the time at which the requests are processed.

We start with the time required for single file encryption. The time is only the time elapsed in process waiting queue of the TPM server in addition to encryption time. Later we repeat the process and find a generalized average time of uploading of a single file. Attempt Count on x-axis shows the number of times the readings were taken. Y-axis denotes the time taken in encryption (Fig. 3).

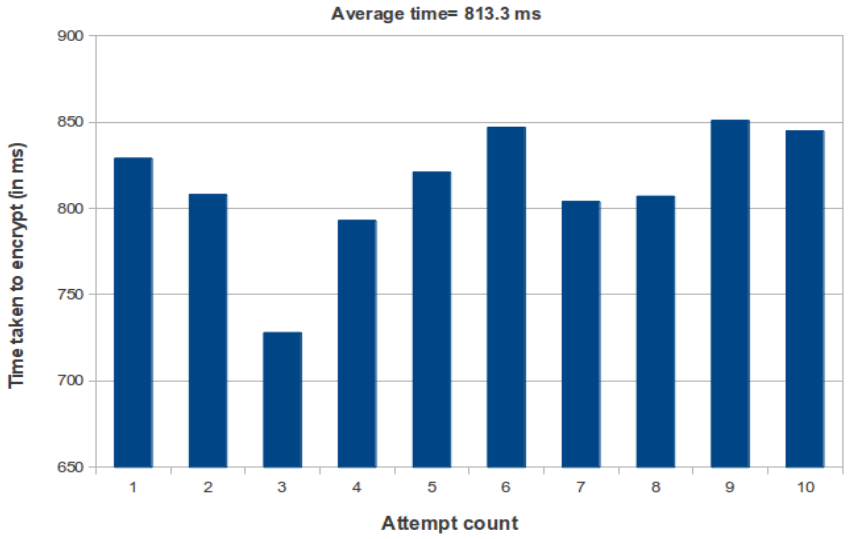

Fig. 3 Encryption time for single file, iterated n= 10 times

It is interesting to see that the time required for encryption varies irrespective of the fact that a single file was uploaded. This phenomenon occurs due to presence of other process at the TPM server. Single file uploading signifies the uploading of a single file from our test system. There may be other systems/clients who are also requesting for the server. Number of such processes (from other then test system) is randomized. To freeze to an acceptable time, we average out the times taken at different iterations. This gives us a file encryption time for single file.

The similar process is repeated for uploading of 50, 100, 150 and 200 files. This is done to test the impact of stress on the TPM server. (Shown in Fig. 4)

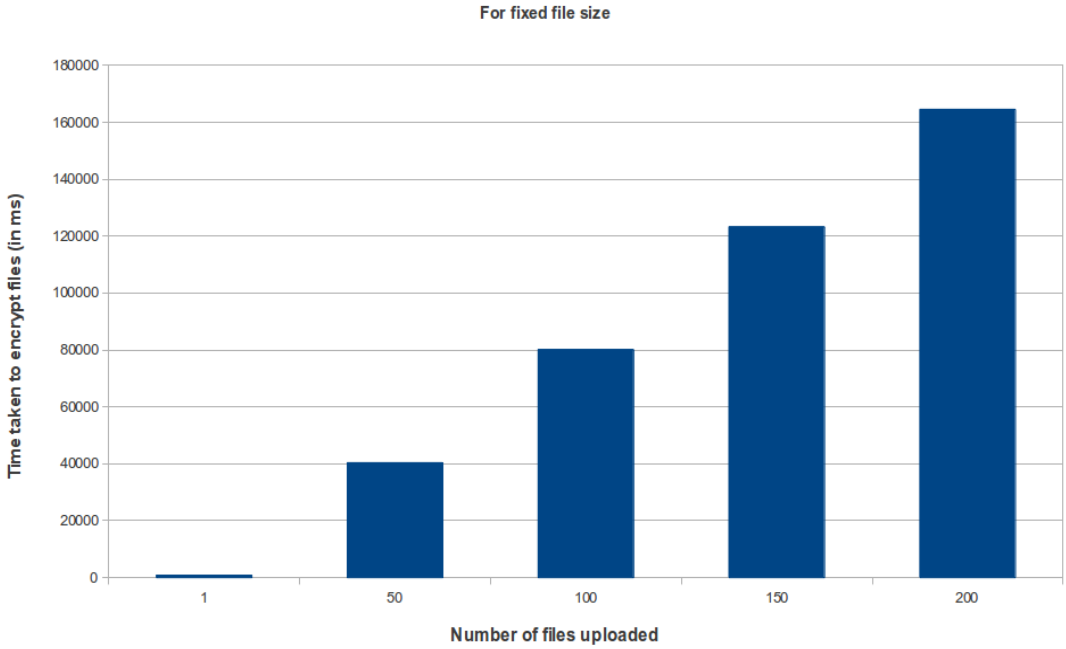

Fig. 4 Encryption time for varying no. of files of same size

The above figure represents a unique feature of the proposed system. The example below testifies it further. Ex: Suppose the time taken for the encryption of single file is 813.3msec (taken from graph). So, the expected time of encryption for 50 files must be 813.3*50 = 40665 msec (since the execution is sequential in TPM), which is very close to observed reading of 40368.6 msec for 50 files. (as in Fig. 4)

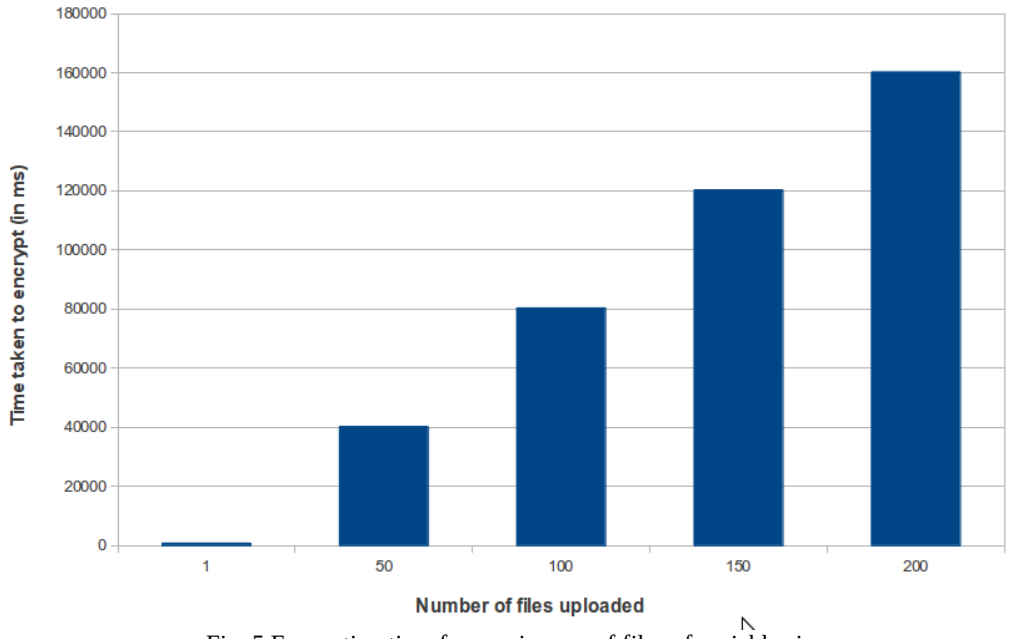

Fig. 5 Encryption time for varying no. of files of variable size

The above argument is valid only when considered for files of similar size. But the proposed system is also evaluated with randomly generated file sizes for the files i.e. size of all 50 files is not same (Fig. 5). Under such diversity, system performs in equal to a standard system of equal sizes, which definitely is an improvement.

VI. CONCLUSIONS

We have implemented a model which helps to use trusted platform module widely for the security of cloud/remote storage. We have design a new trust model which uses TPM to store encrypted data to the cloud, which is unprofitable to the other users. The data will be safe in the public cloud also since one would need the same TPM to get the keys to decrypt the data on cloud. In TPM access to keys, data or systems is often protected (i.e. keys are stores in the shielded location protected by another key) and requires authentication too to access them.

VII. FUTURE WORK

The architecture uses to secure cryptographic keys using TPM and it provides highest level of security to the keys encrypted using the TPM. Here we have given emphasize only on the deployment on TPM in the proposed architecture. In near future we can enhance the implementation of the architecture with a proper key management and we can make the architecture more secure by installing firewalls and others security mechanisms.

REFERENCES

[1] Trusted Computing Group. [Online].
Available: https://www.trustedcomputinggroup.org/
[2] William Stallings, Cryptography and Network Security- Principles and Practices, 3rd Edition, Prentice Hall of India, 2003.
[3] TPM/J Java-based API for the Trusted Platform Module (TPM). [online].
Available: http://projects.csail.mit.edu/tc/tpmj/
[4] Abhishek Patel and Mayank Kumar, "A Proposed Model for Data Security of Cloud Storage Using Trusted Platform Module," International Journal of Advanced Research in Computer Science and Software Engineering on, Volume 3, Issue 4, April 2013, pp. 862-866, ISSN: 2277 128X.
[5] K.Valli Madhavi, R. Tamilkodi and R.Bala Dinakar, "Data Storage Security in Cloud Computing for Ensuring Effective and Flexible Distributed System," International Journal of Electronics Communicationand Computer Engineering, Volume 3,Issue (1) NCRTCST, ISSN 2249 –071X.
[6] Shyam Patidar, Dheeraj Rane and Pritesh Jain, "A Survey Paper on Cloud Computing," 2012 Second

International Conference on Advanced Computing & Communication Technologies."

[7] Kailash Patidar, Ravindra Gupta, Gajendra Singh, Megha Jain and Priyanka Shrivastava, "Integrating the Trusted Computing Platform into the Security of Cloud Computing System," International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 2, Issue 2, February 2012.

[8] Gurudatt Kulkarni, Ramesh Sutar and Jayant Gambhir, "Cloud Computing-Storage as Service," International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, Vol. 2, Issue 1,Jan-Feb 2012, pp.945-950.

[9] S.Sajithabanu and E.George Prakash Raj, "Data Storage Security in Cloud," InternatIonal Journal of Computer Science and Technology, ISSN: 0976-8491 (Online)| ISSN: 2229-4333 (Print), IJCST Vol. 2, Issue 4, oct. -Dec. 2011.

[10] Hari Baaskar R and Gomathi A, "A Framework for Security Based Cloud by using Trusted Computing," International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 2, Issue 12, December 2012.

[11] Nashaat el-Khameesy and Hossam Abdel Rahman, "A Proposed Model for Enhancing Data Storage Security in Cloud Computing Systems," Journal of Emerging Trends in Computing and Information Sciences, ISSN 2079-8407, Vol. 3, Nn. 6, June 2012.

[12] Cong Wang, Qian Wang, KuiRen, Ning Cao and Wenjing Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," Services Computing, IEEE Transactions on , vol.5, no.2, pp.220,232, April-June 2012.

