International Conference on Information Engineering, Management and Security
2015 [ICIEMS 2015]

ISBN	978-81-929742-7-9	**VOL**	01
Website	www.iciems.in	**eMail**	iciems@asdf.res.in
Received	10 - July - 2015	**Accepted**	31- July - 2015
Article ID	ICIEMS011	**eAID**	ICIEMS.2015.011

CLOUD DATA PROTECTION FOR THE MASSESS

V.Prasanth[1], B.Ajay[2], R.Nijanthan[3]
[1]Dept. of computer science and engineering, Velammal Institute of technology,Chennai
[2]Dept. of computer science and engineering, S.R.M. University,Chennai
[3]Dept. of computer science and engineering, Velammal Institute of technology,Chennai,

Abstract: *Offering strong data protection to cloud users while enabling rich applications is a challenging task. We explore a new cloud platform architecture called Data Protection as a Service, which dramatically reduces the per-application development effort required to offer data protection, while still allowing rapid development and maintenance.*

Intoduction

Although cloud computing promises lower costs, rapid scaling, easier maintenance, and service availability anywhere, anytime, a key challenge is how to ensure and build confidence that the cloud can handle user data securely. A recent Microsoft survey found that —58 percent of the public and 86 per-cent of business leaders are excited about the possibilities of cloud computing. But more than 90 percent of them are worried about security, availability, and privacy of their data as it rests in the cloud.‖1 This tension makes sense: users want to maintain con-trol of their data, but they also want to benefit from the rich services that application developers can provide using that data. So far, the cloud offers little platform-level support or standardization for user data protection beyond data en-cryption at rest, most likely because doing so is nontrivial.
Protecting user data while enabling rich computation re-quires both specialized expertise and resources that might not be readily available to most application developers.
Building in data-protection solutions at the platform layer is an attractive option: the platform can achieve economies of scale by amortizing expertise costs and dis-tributing sophisticated security solutions across different applications and their developers.

WHAT ABOUT ENCRYPTION?

In the realm of data protection, *full-disk encryption* (FDE) and computing on encrypted data have recently gained attention, these techniques have fallen short of answering all of the security and maintenance challenges mentioned earlier.FDE encrypts entire physical disks with a symmetric key, often in disk firmware, for simplicity and speed. At the other end of the spectrum, Craig Gentry re- cently proposed the first realization of *fully homomorphic encryption* (FHE),[2] which offers the promise of general computation on ciphertexts. Basically, any function in plaintext can be transformed into an equivalent function in ciphertext: the server does the real work, but it doesŶ't kŶow the data it's ĐoŵputiŶg. Naturally, this property gives strong privacy guarantees when computing on private data, but the question of its practicality for general cloud applications still remains.

Cite this article as: V.Prasanth, B.Ajay, R.Nijanthan. "CLOUD DATA PROTECTION FOR THE MASSESS." *International Conference on Information Engineering, Management and Security (2015)*: 70-73. Print.

1.FDE versus FHE

A comparison of FDE and FHE in the cloud comput-ing setting reveals how these encryption techniques fall short of addressing the aforementioned security and maintenance challenges simultaneously.

1.1 Key management and trust.

With FDE, the keys reside with the cloud platform, generally on or close to the physic cal drive: the cloud application user isn't involved in key management. While user data is encrypted on the physical disk, it is always accessible in the clear to any layer above it. Consequently, FDE doesn't prevent online attacks from leaking the data to an unauthorized party, which is far more common in the cloud setting than physical attacks. With FHE, untrusted applications can't easily learn or leak data. Users typically own and manage FHE encryp-tion keys, while applications compute on encrypted forms of user data without actually seeing‖ the data. This raises questions about how users can store their keys securely and reliably, especially in the presence of sharing. After all, the point of the cloud is to avoid maintaining local state.

1.2 Sharing.

Collaboration is often cited as a —killer feature‖ for cloud applications. Fine-grained access control is nec-essary to let a data owner selectively share one or more data objects with other users.

With FDE, users must fully trust the cloud provider to enforce correct access control because the key granularity (the whole disk) doesn't line up with access control granu-larity (a single data unit).

With FHE, because the user—or a third-party cloud pro-vider employed by the user—manages the encryption keys, the best way of providing access control isn't clear yet. To offer fine-grained encryption-based access control, we might need to define key management on a per data object granularity basis or over collections of data objects. How-ever, to support homomorphic operations across multiple encrypted objects, those objects must still be encrypted under the same public key.

1.3 Aggregation.

Many cloud applications require per-forming data mining over multiple users' data for tasks such as spam filtering or computing aggregate statistics. Because users fully trust the cloud provider, performing such data aggregation is relatively easy with FDE.

Current FHE techniques don't readily allow computing on multiple users' data encrypted under different keys. Therefore, it isn't clear yet how to support such data aggre-gation applications with FHE; similarly, offline aggregation across users' data isn't possible. One solution might be to escrow keys to the cloud provider, but that would eliminate many of FHE's benefits, making its cost harder to justify.

1.4 Performance.

According to a recent survey, 49 percent of users abandon a site or switch to a competitor after e xperiencing performance issues.[3] And the need for speed is only increasing: in 2000, a typical user was willing to wait 8 seconds for a webpage to load before navigating away; by 2009, that number dropped to 3 seconds.

When FDE is implemented in disk firmware, its sym-metric encryption can run at the disk's full bandwidth, effectively avoiding a slowdown. Although researchers have made significant advances in improving FHE's per-formance since Gentry's original proposal, it has a long way to go before becoming efficient enough to deploy at scale. In Gentry's estimation, implementing something like a Google search with FHE would require roughly 1 trillion times more computation than the one without FHE.[4]

1.5 Ease of development.

Because FDE is hidden behind an abstraction of the physical disk, it typically has no impact on application development. In theory, FHE could also be relatively automatic: it works on an abstraction of the program as a circuit and transforms that circuit. In practice, however, performing this translation for arbitrary programs—especially when marshaling data could be quite complex. At a minimum, programming tools would need to evolve dramatically.

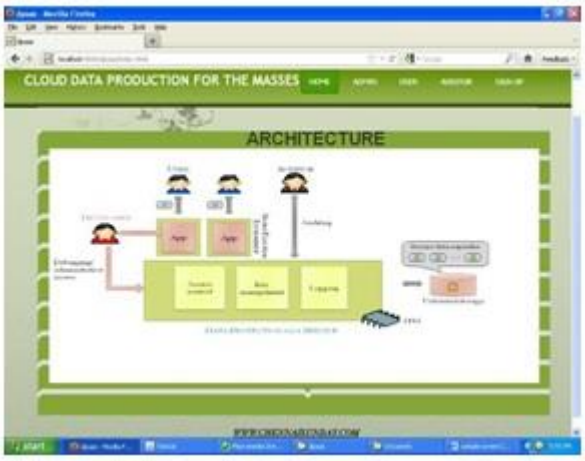
2. Splitting the difference

Although FDE offers e xcellent performance and ease of development, it does little to protect privacy at the required granularity. FHE, on the other hand, pushes the privacy envelope in the other direction by removing data visibility entirely from both the server and application developer. However, having a remote machine see and compute on sensitive data isn't automatically a privacy violation. FHE's guarantees go beyond what's necessary to protect data, and in so doing, it incurs significant performance and de-velopment costs. We believe the DPaaS approach is better suited for the target applications because it falls between the two. It keeps the —natural‖ granularity of FHE by keying on units of sharable data and maintains the performance of FDE by using symmetric encryption. It moves key manage-ment and access control to a middle tier—the computing platform—to balance rapid development and easy main-tenance with user- side verifiability.

Cite this article as: V.Prasanth, B.Ajay, R.Nijanthan. "CLOUD DATA PROTECTION FOR THE MASSESS." *International Conference on Information Engineering, Management and Security (2015)*: 70-73. Print.

A WAY FORWARD

In an OS, processes and files are the primary units of access control, and the OS provides suitable isolation for these boundaries.



In a cloud setting, the unit of access control is typically a sharable piece of user data—for example, a document in a collaborative editor. Ideally, the system offers some analogous confinement of that data, restricting its visibility only to authorized users and applications while allowing broad latitude for what operations are done on it. This can make writing secure systems easier for programmers because confinement makes it more difficult for buggy code to leak data or for compromised code to grant unau- thorized access to data. A malicious program might find different ways to exfiltrate data, such as employing a side channel or covert channel, but the priority here is to sup-port benign developers, while making all applications and their actions on users' sensitive data more easily auditable to catch improper usage.

One of the main concerns people and organizations have about putting data in the cloud is that they don't know what happens to it. Having a clear audit trail of when data is accessed—and by whom or what— bolsters confidence that data is being handled appropriately. Con-finement can be effective for most normal user accesses, but administrative access that's outside the normal flow of user access and involves human administrators (for example, for debugging and analysis) can especially ben-efit from auditing.

3. Verifiable platform support

Bugs need to be fixed. Data needs to be updated and amigrated as schemas change. Offline computation is valuable for data aggregation across users or for precom-putation of expensive functions. To reduce the risk of unaudited backdoor access, all these functions should be subject to the same authorization flows and platform-level checks as normal requests, albeit with a separate, appropriate policy.

Platform providers should build support for confine-ment and auditing into the platform in a verifiable way. This approval has many advantages:

- application developers don't have to reinvent the wheel;
- application code is independent of ACL enforcement;
- third-party auditing and standards compliance are easier; and
- the verifiable platform extends to virtualized environ-ments built atop it.
- Finally, the cost of examining the platform is amortized across all its users, which means significant economies of scale.

4. Design space and a sample architecture

Figure 1 illustrates an example architecture for explor-ing the DPaaS design space.[5] Here, each server contains a *trusted platform module* (TPM) to provide secure and verifiable boot and dynamic root of trust. This example ar-chitecture demonstrates at a high level how it's potentially possible to combine various technologies such as applica-tion confinement, encryption, logging, code attestation, and information flow checking to realize DPaaS.

4.1 Confinement.

A *secure data capsule* (SDC) is an encrypted data unit packaged with its security policy. For example, an SDC might encompass a sharable docu-ment or a photo album along with its ACL. The platform can use confinement and information-flow controls to enforce capsules' ACLs.

4.2 Audit trails.

Because the platform mediates all data access, authenticates users, and runs binaries, it knows what data is accessed by what user, and with which appli-cation. It can generate meaningful audit logs containing all these parameters and optionally incorporate additional information from the application layer.

Cite this article as: V.Prasanth, B.Ajay, R.Nijanthan. "CLOUD DATA PROTECTION FOR THE MASSESS." *International Conference on Information Engineering, Management and Security (2015)*: 70-73. Print.

4.3 Platform verifiability.

The DPaaS approach provides logging and auditing at the platform level, sharing the benefits with all applications running on top. Offline, the auditor can verify that the platform implements each data protection feature as promised. At runtime, the platform provider can use *trusted computing* (TC) technologies to attest to the particular software that's running. TC uses the tamperproof TPM as well as the virtualization and isolation features of modern processors, such as Intel VT or AMDV.

5.Achieving data protection goals

We assume in the analysis that the platform behaves correctly with respect to code loading, authorization, and key management, and that the TPM facilitates a runtime attestation to this effect. DPaas uses a combination of encryption at rest, ap-plication confinement, information flow checking, and auditing to ensure the security and privacy of users' data. Application confinement isolates faults and compromises within each SEE, while information flow checking ensures that any information flowing among SEEs, data capsules, and users satisfies access-control policies. Controlling and auditing administrative accesses to data provides account-ability. DPaaS can guarantee the integrity of the data at rest via cryptographic authentication of the data in storage and by auditing the application code at runtime.

Access controls, authorization, and auditing capability are common challenges for application developers. Incor-porating these features within the platform is a significant improvement in terms of ease of use, and it doesn't con-strain the types of computation that can be performed within a SEE. The platform logs common maintenance and batch processing tasks to provide accountability. These tasks too often require one-off work in the development process and can benefit from standardization.

CONCLUSION:

As private data moves online, the need to secure it properly becomes increasingly urgent. The good news is that the same forces concentrating data in enormous datacenters will also aid in using collective security expertise more effectively. Adding protections to a single cloud platform can immediately benefit hundreds of thousands of applications and, by extension, hundreds of millions of users. While we have focused here on a particular, albeit popular and privacy-sensitive, classes of applications, many other applications also need solutions.

References

1. C. Dwork,—The Differential Privacy Frontier Extended Abstract,‖ *Proc. 6th Theory of Cryptography Conf.* (TCC 09), LNCS 5444, Springer, 2009, pp. 496-502.
2. C. Gentry,—Fully Homomorphic Encryption Using Ideal Lattices,‖ *Proc. 41st Ann. ACM Symp. Theory Computing* (STOC 09), ACM, 2009, pp. 169-178.
3. E. Naone,—The Slow-Motion Internet,‖ *Technology Rev.*,Mar./Apr. 2011; www.technologyreview.com/files/54902/Google Speed_charts.pdf.

Cite this article as: V.Prasanth, B.Ajay, R.Nijanthan. "CLOUD DATA PROTECTION FOR THE MASSESS." *International Conference on Information Engineering, Management and Security (2015)*: 70-73. Print.