Data Protection for Cloud Users

Prof. R.T Nakhate[*], Prof. B.B Lonkar[**]

[*] Nagpur University, DMIETR, Salod Wardha.
[**] Nagpur University, DMIETR, Salod Wardha

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.

Index Terms- data, security, cloud, disk.

I. INTRODUCTION

Cloud computing promises lower costs, rapid scaling, easier maintenance, and services that are available anywhere, anytime. A key challenge in moving to the cloud is to ensure and build confidence that user data is handled securely in the cloud. A recent Microsoft survey [10] found that "...58% of the public and 86% of business leaders are excited about the possibilities of cloud computing. But, more than 90% of them are worried about security, availability, and privacy of their data as it rests in the cloud."

A. Data Protection and Usability Properties

Integrity: The user's private (including shared) data is stored faithfully, and will not be corrupted.

Privacy: The user's private data will not be leaked to any unauthorized person.

Access transparency: It should be possible to obtain a log of accesses to data indicating who or what performed each access.

Ease of verification: It should be possible to offer some level of transparency to the users, such that they can to some extent verify what platform or application code is running. Users may also wish to verify that their privacy policies have been strictly enforced by the cloud.

Rich computation: The platform allows most computations on sensitive user data, and can run those computations efficiently.

Development and maintenance support: Any developer faces a long list of challenges: bugs to find and fix, frequent software upgrades, continuous change of usage patterns, and users' demand for high performance. Any credible data protection approach must grapple with these issues, which are often overlooked in the literature on the topic.

B. Target Applications

There is a real danger in trying to "solve security and privacy for the cloud," because "the cloud" means too many different things to admit any one solution. To make any actionable statements, we must constrain ourselves to a particular domain. We choose to focus on an important class of widely used applications which includes email, personal financial management, social networks, and business applications such as word processors and spreadsheets. More precisely, we focus on deployments which meet the following criteria:

• Applications that provide services to a large number of distinct end users, as opposed to bulk data processing or workflow management for a single entity;

• Applications whose data model consists mostly of sharable data units, where all data objects have ACLs consisting of one or more end users (or may be designated as public); and developers who write applications to run on a separate computing platform—which encompasses the physical infrastructure, job scheduling, user authentication, and the base software environment—rather than implementing the platform themselves.

C DATA Protection as a Service

Currently, users have to rely primarily on legal agreements and implied economic and reputational harms as a proxy for applications' trustworthiness. Ideally, we would like a robust technological solution as the base. To achieve this, we propose that the two most important things that a cloud platform could do are to:

• make it easy for developers to write per formant, maintainable applications that protect user data in the cloud, thereby providing the same economies of scale to security and privacy as for computation and storage;

• enable independent verification both of the platform's operation and the runtime state of applications on it, so users (perhaps directly, but more likely via third-party auditors) can gain confidence that their data is being handled properly.

In the realm of data protection, people often view encryption as a kind of a silver bullet. In reality, encryption is just a tool—albeit a powerful one—to help achieve data protection properties, and not an end in itself. We will discuss two techniques that have received a lot of attention, full-disk encryption and computing on encrypted data, and show how they fall short of achieving our goals.

II. DOES ENCRYPTION SOLVE ALL OUR PROBLEMS?

A. Full-disk Encryption and Computation on Encrypted Data.

To motivate our ultimate proposal, we consider two different approaches to data privacy, full-disk encryption and computation on encrypted data. Full-disk encryption (FDE) refers to encrypting entire physical disks with a symmetric key, often in disk firmware for simplicity and speed. Many standards call for encryption of data at rest, which FDE nominally fulfills. FDE is effective in protecting private data in certain scenarios such as stolen laptops and backup tapes. But does it fulfill our cloud data protection goals in the cloud, where physical theft is not the main threat?

At the other end of the spectrum, modern cryptography offers rich capabilities for computation on encrypted data which were not previously possible. Recently, the first realization of fully homomorphic encryption (FHE) was discovered by Craig Gentry [4]. FHE offers the promise of general computation on ciphertexts. That is, you can take any function on plaintext and transform it into an equivalent function on ciphertext: the server does all the real work, but does not know what the data it is computing on actually is. Naturally, this property gives strong privacy guarantees when computing on private data, but the question of its practicality for general cloud applications still remains.

III. A Way Forward: Data Protection as a Service

Lightweight confinement of user data: In an operating system, processes and files are the primary units of access control, and the OS provides suitable isolation for them. Applications are free to do what they like within these boundaries.

In a cloud setting, the unit of access control is typically a sharable piece of user data (e.g.,a document in a collaborative editor). We would like 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, since confinement makes it more difficult for buggy code to leak data or compromised code to give unauthorized access to data. A malicious program may find different ways to exfiltrate data such as employing a side channel or covert channel; our higher priority here is to support benign developers, while making all applications and their actions on users' sensitive data more easily auditable to catch improper usage.

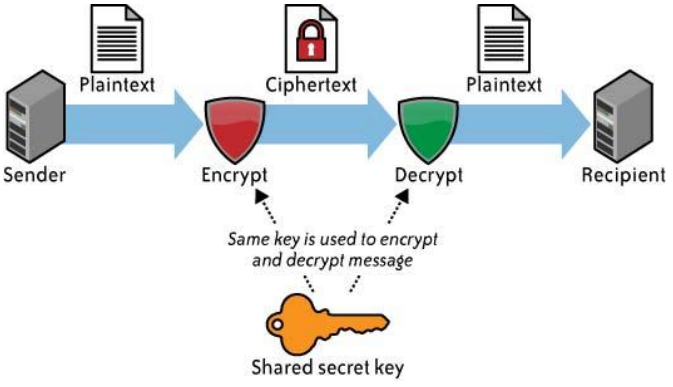

Figure 1 the Encryption Process

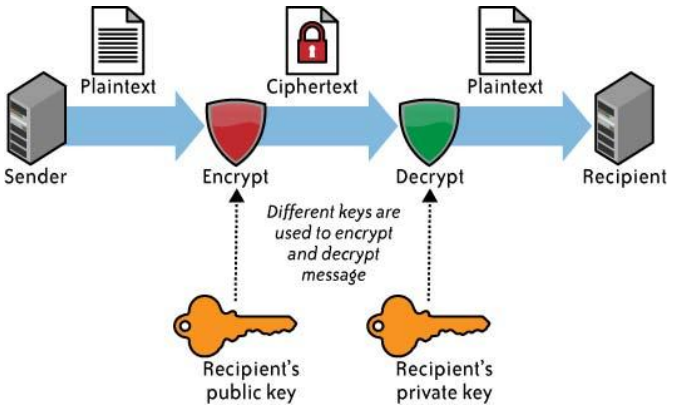

Figure 2 The Decryption Process

Secure support for debugging, maintenance, and batch processing:

These are essential for the proper functioning of virtually all useful applications. Bugs need to be fixed. Data need to be updated and migrated as schemas change. Offline computation is valuable for doing aggregation across users or for precomputation of expensive functions. To reduce the risk of unaudited back-door access, all these should be subject to the same authorization flows and platform-level checks as normal requests, albeit with a separate, appropriate policy.

Verifiable platform-level support:

Support for confinement and auditing should be built into the platform in a verifiable way. This has many advantages: application developers do not all have to reinvent the wheel; the controls maintain independence from application code; third-party auditing and standards compliance are easier; and it allows for hardware support even in a virtualized environment. The cost of examining the platform is amortized across all its users; for a large-scale platform provider, this means significant economies of scale.

A. Data Access Auditing Support

Since the platform mediates all access to the data, authenticates users, and runs binaries, it knows what data is being accessed, by what user, and using which application. It can generate meaningful audit logs containing all these parameters and optionally incorporate additional information from the application layer. There are four basic kinds of actions we can log:

- Ordinary online data accesses occur in response to external requests from users, and take place when a user is online and using an application.
- Access control modification by authorized users. Knowing the provenance of these changes can be helpful for forensics or diagnosing sharing problems.
- Offline/batch access to handle requests while users are offline (e.g., e-mail delivery), to compute aggregates, or to reorganize data such as during schema changes.
- Administrative access for maintenance operations such as debugging.

B. Analysis against Our Data Protection Goals

We assume in this analysis that the platform has been verified to behave correctly with respect to code loading, authorization, and key management, and that there exists a runtime attestation to this effect (made possible by the TPM).

Data protection properties Privacy is ensured by the encryption at rest, and a combination of application confinement and information flow checking. Application confinement isolates faults and compromises within each SEE, while information flow checking ensures that any information flowing amongst SEEs, data capsules, and end users satisfy access control policies. Administrative accesses to data are controlled and audited to provide accountability. Integrity of the data at rest can be obtained by cryptographic authentication of the data in storage, and by audit of the application code at runtime.

Ease of development Access controls, authorization, and auditing capability are common pain points for application developers. Getting these "for free" with the platform is a real improvement in ease of use, and we do not constrain the types of computation that can be performed within an SEE.

Common maintenance tasks and batch processing are provided directly as first-class operations and logged suitably. These too often require one-off work in the development process and can benefit From standardization.

IV. CONCLUSION

As people's private data moves online, the need to secure it properly becomes ever more urgent. We can rely neither on the physical boundaries of the pre-Internet age, nor, with the cloud, the implicit Protection afforded by data being distributed among individual users' machines. Implementing data Protection for a cloud application is hard, requiring specialized skills and significant implementation effort. The good news is that the same forces which are concentrating data in enormous datacenters are also the ones which will let us use our 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. Towards this goal, we have proposed a new paradigm for cloud computing,

Data Protection as a Service, and discussed its key principles and design space. In this paradigm, the cloud platform not only provides the hardware and software stack as in today's cloud computing, but also dynamic data protection that protects users' data while enabling rich computation over them. We have focused here on a particular, albeit popular and privacy-sensitive, class of applications, and shown how to protect its data at the platform level. Many other types of applications also need solutions, and many practical questions still remain open.

We pose the following challenges:

• Can we standardize the technology across platforms, so switching between different providers is easy?

• How can we make migration for existing applications as easy as possible?

• How can we minimize the cost of application audits? What kinds of audits are most important to build users' confidence?

• Can technologies such as Trusted Computing and code attestation be made scalable in the presence of constantly evolving software?

• How can we generalize the ideas here to other classes of applications?

REFERENCES

[1] http://www.mydatacontrol.com.

[2] Theneedforspeed.charts.pdf.http://www.technologyreview.com/files/54902/GoogleSpeed

[3] C. Dwork. The differential privacy frontier. In TCC, 2009.

[4] C. Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In STOC, pages 169–178,2009.

[5] A. Greenberg. IBM's Blindfolded Calculator. Forbes, June 2009. Appeared in the July 13,2009 issue of Forbes magazine.

[6] P. Maniatis, D. Akhawe, K. Fall, E. Shi, S. McCamant, and D. Song. Do You Know WhereYour Data Are? Secure Data Capsules for Deployable Data Protection. In HotOS, 2011.

[7] S. McCamant and M. D. Ernst. Quantitative information flow as network flow capacity. InPLDI, pages 193–205, 2008.

[8] M. S. Miller. Towards a Unified Approach to Access Control and Concurrency Control. PhDthesis, Johns Hopkins University, Baltimore, Maryland, USA, May 2006.

[9] A. Sabelfeld and A. C. Myers. Language-Based Information-Flow Security. IEEE Journal onSelected Areas in Communications, 21(1):5–19, 2003.

[10] L. Whitney. Microsoft Urges Laws to Boost Trust in the Cloud. http://news.cnet.com/8301-1009 3-10437844-83.html.

[11] Cloud Security and privacy by Tim Mather, Subra kamarasaway , Shahed Latif.

AUTHORS

First Author – Prof. R.T Nakhate, Nagpur University, DMIETR, Salod Wardha., Rajesh.nakhate@gmail.com
Second Author – Prof. B.B Lonkar, Nagpur University, DMIETR, Salod Wardha., bhu_lon@yahoo.com