Research Paper Engineering

Security For Bulk Data In Cloud

* K. Uday kumar ** S. K. Prashanth

* M.TECH (SE), VCE, Hyderabad, India.

** Prof .CSE, VCE, Hyderabad, India.

ABSTRACT

Offering strong data protection to cloud users whileenabling 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.

Keywords : Rural water supply, Level of Service, Supply driven approach, Willing to pay

Introduction:
This tension makes sense: users want to maintain control 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 encryption at rest, mostlikely because doing so is nontrivial. Protecting user data while enabling rich computation requires 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 distributing sophisticatedsecurity solutions across different applications and their developers.

Wepropose a new cloud computing paradigm, data protection as a service (www.mydatacontrol.com). DPaaS is a suite of security primitives offered by a cloud platform, which enforces data security and privacy and offers evidence of privacy to data owners, even in the presence of potentially compromised or malicious applications.

SECURITY AND PRIVACY CHALLENGES
It's impossible to develop a single data- protection solution for the cloud because the term means too many different things. Any progress must first occur in a particular domain accordingly, our work focuses on an important class of widely used applications that includes e-mail, personal financial management, social net works, and business tools such as word processors and spreadsheets.

The following criteria define this class of applications:
COVATURE

- provide services to a large number of distinct end users, as opposed to bulk data processing or workflow management for a single entity;
- use a data model consisting mostly of sharable units, where all data objects have access control lists (ACLs) with one or more users; and
- Developers could run the applications on a separate computing platform that encompasses the physical infrastructure, job scheduling, user authentication, and the base software environment, rather than implementing the platform themselves.

Overly rigid security is as detrimental to cloud service value as inadequate security.

A primary challenge in designing a platform layer solution useful to many applications is ensuring that it enables rapid development and maintenance. To ensure a practical solution, we considered the following goals relating to data protection as well as ease of development and maintenance:

- Integrity. The user's stored data won't be corrupted.
- Privacy. Private data won't be leaked to any unauthorized entity.
- Access transparency. Logs will clearly indicate who or what accessed any data.
- Ease of verification. Users will be able to easily verify what platform or application code is running, as well as whether the cloud has strictly enforced their data's privacy policies.

process, cloud platforms could offer transparently verifiable partitions for applications that compute on data units, while still allowing broad computational latitude within those partitions.

DPaaS enforces fine-grained access control policies on data units through application confinement and information flow checking. It employs cryptographic protections at rest and offers robust logging and auditing to provide accountability. Crucially, DPaaS also directly addresses the issues of rapid development and maintenance.

To truly support this vision, cloud platform providers would have to offer DPaaSin addition to their existing hosting environment, which could be especially beneficial for small companies or developers who don't have much in-house security expertise, helping them build user confidence much more quickly than they otherwise might.

WHAT ABOUT ENCRYPTION?
In the realm of data protection,developers often view encryption as a kind of a silver bullet, but in reality, it's just a tool albeit a powerful one to help achieve data protection properties.

Although 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.

Although FDE is effective in protecting private data in certain such as stolen laptops and backup tapes, the concern is that it can't fulfill data protection goals in the cloud, where physical theft isn't the main threat.

At the o t h e r end of the spectrum, Craig Gentry recently proposed the first realization of fully homomorphism encryption (FHE),2 which offers the promise of general computation on cipher texts. Basically, any function in plaintext can be transformed into an equivalent function in cipher text: the server does the real work, but it doesn't know the data it's computing. Naturally, this property gives strong privacy guarantees when computing on private data, but the question of its practicality for general cloud applications still remains.

FDE versus FHE

A comparison of FDE and FHE in the cloud computing setting reveals how these encryption techniques fall short of addressing the a fore mentioned security and maintenance challenges simultaneously.

Key management and trust. With FDE, the keys reside with the cloud platform, generally on or close to the physical 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 Users or developers can decide how detailed the logs are on a case- by-case basis Sharing. Collaboration is often cited as a "killer feature" for cloud applications. Fine-grained access control is necessary to let a data owner selectively share one or more data objects with other users. Maintenance. Bugs are inevitable. However, availability is a primary cloud goal, so the need to debug quickly is a top priority.

Given its ability to perform different types of audit, DPaaS can also support third-party auditing services, thus helping users understand how their data has been accessed and manipulated, and which services to trust. We anticipate that auditors will provide personalized services to particular users, helping them determine how safe their data is with a particular service.

The ACL governs ordinary user access, but administrative access requires its own separate policy, which in turn can be audited to hold developers and administrators accountable. Because each specific invocation of the administrative policy might entail human access to data, it should be logged and made available for auditing. The same kind of mechanism could handle batch access, perhaps with different logging granularity. To prevent misuse, the platform can restrict batch processes to only an approved set of programs, for example, requiring the programs to have controlled or quantifiable information release, such as differential privacy or quantitative information flow.

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 tamper proof TPM as well as the virtualization and isolation features of modern processors, such as Intel VT or AMDV.

TC also allows for a dynamic root of trust— while the system runs, the CPU can enter a clean state, and the TPM can verify, load. trustedcomputing base (TCB), which is responsible for security-critical functionalities such as isolation enforcement, key management, access control, and logging. Moreover, a third-party auditor can verify the code of the TCB that has been loaded onto the cloud platform. In this way, users and developers can gain confidence that the applications are indeed running on the correct TCB, and consequently trust the security guarantees and the audit logs the TCB provides.

One challenge in code attestation is how to establish a set of acceptable binaries in the presence of rapid software updates such as bug fixes and new features. One potential way is to log the history of software updates and perform verification a posteriori.

For the application itself, getting from verifiable to verified isn't easy; in a system with a lot of users, doing all- pairs verification is prohibitively expensive. This is where auditors come in. Certifications such as Statement on Auditing Standards Number 70 (SAS70) and others serve the important function of reducing the verification burden on both clients and service providers compared to pairwise examinations. Since applications have the data-protection piece in common from the platforms, the application verifications in turn can be simpler than they otherwise would have been.

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, application confinement, information flow checking, and auditing to ensure the security and privacy of users' data application.

Access controls, authorization, and auditing capability are common challenges for application developers. Incorporating 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.

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/ GoogleSpeed_charts.pdf. | | | 4. S. McCamant and M.D. Ernst, "Quantitative | Information Flow as Network Flow Capacity," Proc. | 2008 ACM SIGPLAN Conf. Programming Language Design and Implementation (PLDI 08), ACM, 2008, pp. 193-205. |