Cloud Data Protection and Benefits in IT Demanding Sectors

[1] Mubeen, [2] G Bhargavi

[1]*Department of CSE, Prakasam Engineering College, Kandukuru,Prakasam dist., Ap,India.*

[2] *Associate. Professor Department of IT, Prakasam Engineering College, Kandukuru,Prakasam dist., Ap,India.*

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. 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 percent 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." We propose a new cloud computing paradigm, data protection as a service (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. Such as secure data using encryption, logging, key management.***

Keywords*— Cloud Computing, Cloud Benefits, Cloud Risk.*

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."

There is tension between user data protection and rich computation in the cloud. Users want to maintain control of their data, but also want to benefit from rich services provided by application developers using that data. At present, there is little platform-level support and standardization for verifiable data protection in the cloud. On the other hand, user data protection while enabling rich computation is challenging. It requires specialized expertise and a lot of resources to build, which may not be readily available to most application developers. We argue that it is highly valuable to build in data protection solutions at the platform layer: The platform can be a great place to achieve economy of scale for security, by amortizing the cost of maintaining expertise and building sophisticated security solutions across different applications and their developers.

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

Data Protection and Usability Properties

A primary challenge in designing a platform-layer solution useful to many applications is allowing rapid development and maintenance. Overly rigid security will be as detrimental to cloud services' value as inadequate security. Developers do not want their security problems solved by losing their users! To ensure a practical solution, we consider goals relating to data protection as well as ease of development and maintenance.

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.

Existing System

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 percent 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."

Proposed System

We propose a new cloud computing paradigm, *data protection as a service* (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. Such as secure data using encryption, logging, key management

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.
-

OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on

implementation, designing of methods to achieve changeover and evaluation of changeover methods.

MODULE DESCRIPTION:

1. **Cloud Computing**
2. **Trusted Platform Module**
3. **Third Party Auditor**
4. **User Module**

1. Cloud Computing

Cloud computing is the provision of dynamically scalable and often virtualized resources as a services over the internet Users need not have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them. Cloud computing represents a major change in how we store information and run applications. Instead of hosting apps and data on an individual desktop computer, everything is hosted in the "cloud"—an assemblage of computers and servers accessed via the Internet.

Cloud computing exhibits the following key characteristics:

1. Agility improves with users' ability to re-provision technological infrastructure resources.

2. Multi tenancy enables sharing of resources and costs across a large pool of users thus allowing for:

3. Utilization and efficiency improvements for systems that are often only 10–20% utilized.

4. Reliability is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for business continuity and disaster recovery.

5. Performance is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface.

6. Security could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems that are being shared by unrelated users. In addition, user access to security Audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

7. Maintenanceof cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

2 .Trusted Platform Module

Trusted Platform Module (TPM) is both the name of a published Specification detailing a secure crypto processor that can store cryptographic Keys that protect information, as well as the general name of implementations of that specification, often called the "TPM chip" or "TPM Security Device". The TPM specification is the work of the Trusted Computing Group.

Disk encryption is a technology which protects information by converting it into unreadable code that cannot be deciphered easily by unauthorized people. **Disk encryption** uses disk encryption software or hardware to encryptevery bit of data that goes on a disk or disk volumeDisk encryption prevents unauthorized access to data storage. The term "full disk encryption" (or **whole disk encryption**) is often used to signify that everything on a disk is encrypted, including the programs that can encrypt bootable operating system partitions. But they must still leave the master boot record (MBR), and thus part of the disk, unencrypted. There are, however, hardware-based full disk encryption systems that can truly encrypt the entire boot disk, including the MBR**.**

3. Third Party Auditor

In this module, Auditor views the all user data and verifying data and also changed data. Auditor directly views all user data without key. Admin provided the permission to Auditor. After auditing data, store to the cloud.

4. User Module

User store large amount of data to clouds and access data using secure key. Secure key provided admin after encrypting data. Encrypt the data using TPM. User store data after auditor, view and verifying data and also changed data. User again views data at that time admin provided the message to user only changes data.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific

business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

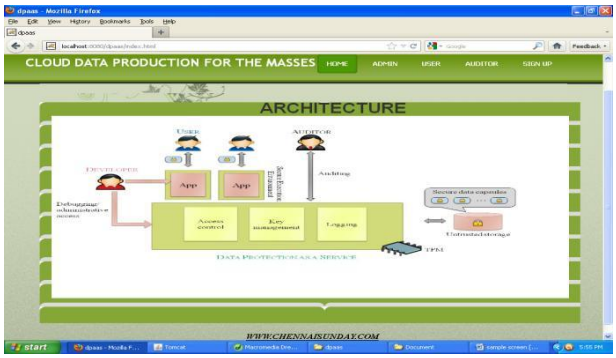
The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

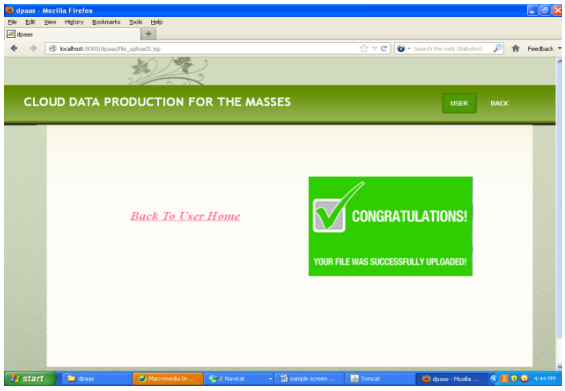
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

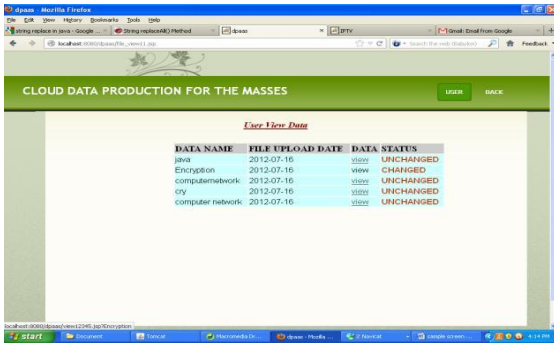
Acceptance Testing

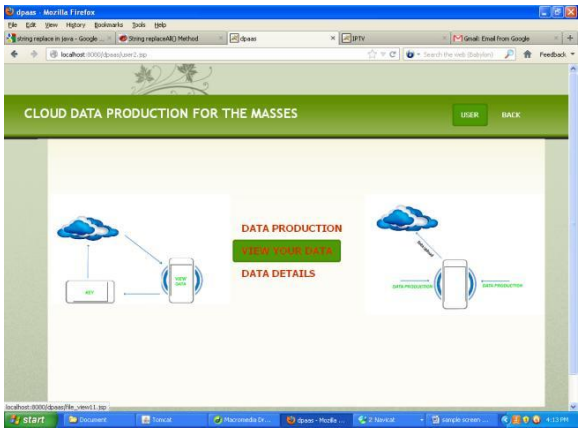
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

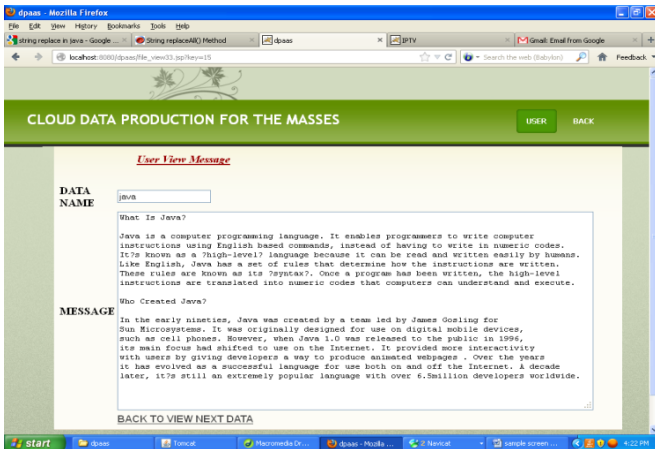
Results All the test cases mentioned above passed successfully. No defects encountered.

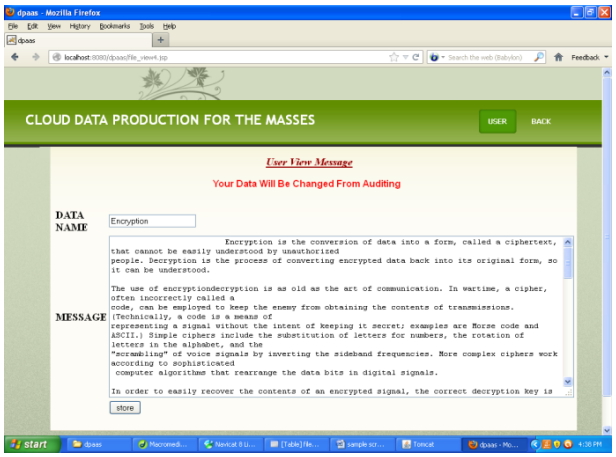


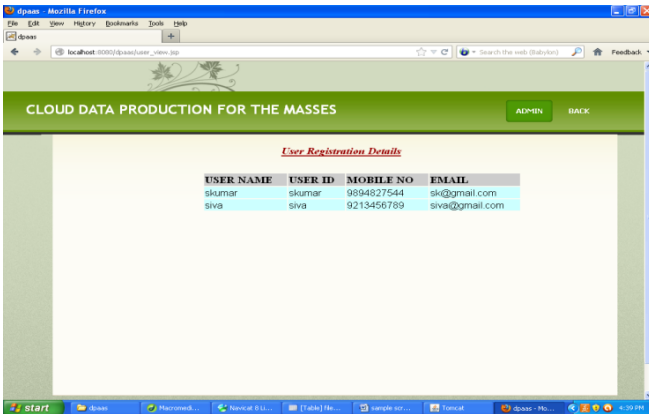


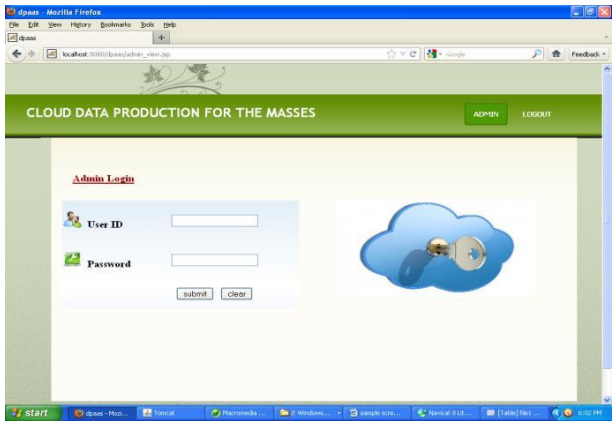


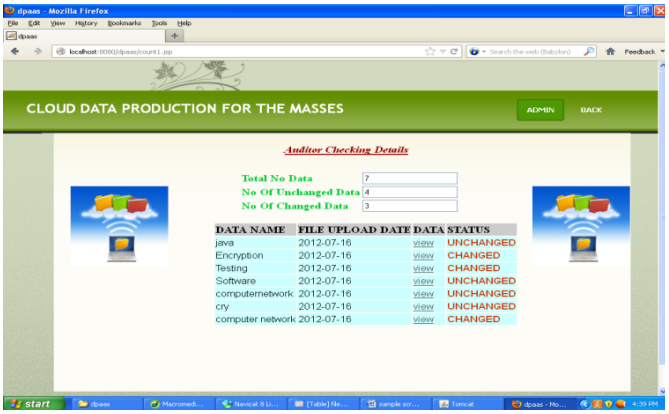


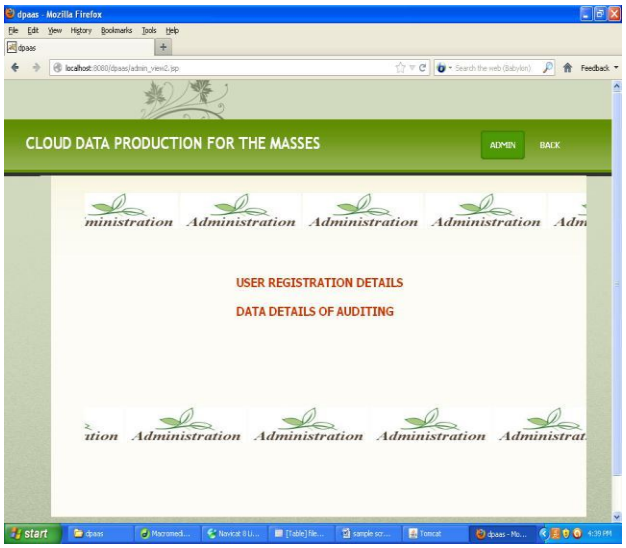


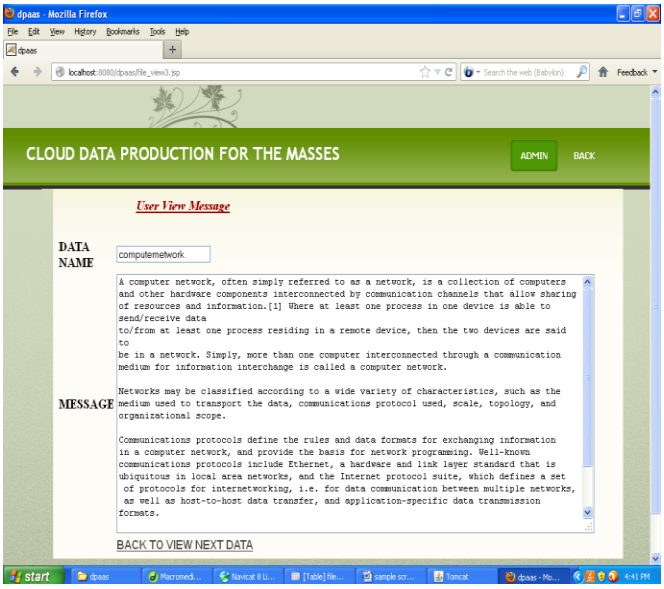


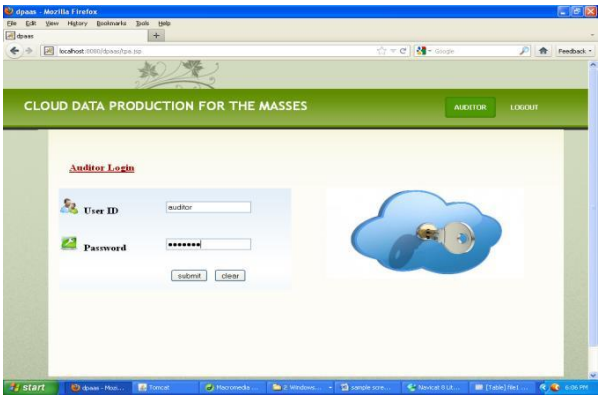


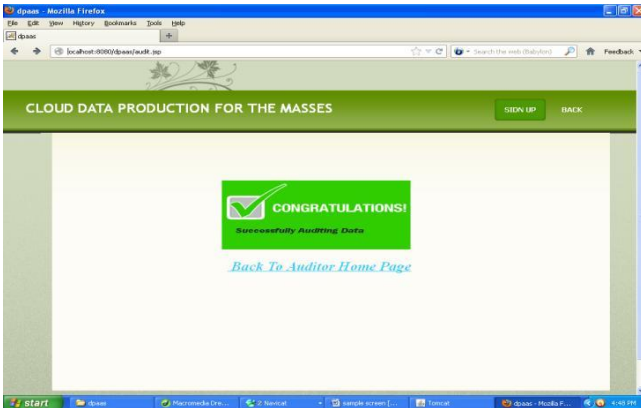


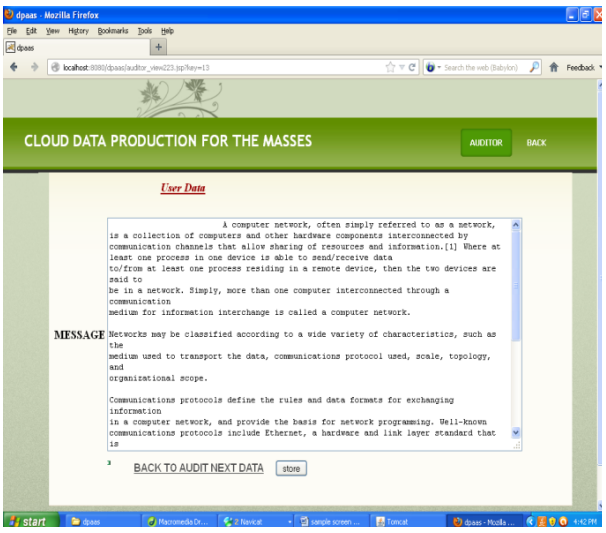












CONCLUSION

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] http://www.mydatacontrol.com.
[2] The need for speed. http://www.technologyreview.com/files/54902/GoogleSpeedcharts.pdf.
[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 Where Your 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. In PLDI, pages 193–205, 2008.
[8] M. S. Miller. Towards a Unified Approach to Access Control and Concurrency Control. PhD thesis, Johns Hopkins University, Baltimore, Maryland, USA, May 2006.
[9] A. Sabelfeld and A. C. Myers. Language-Based Information-Flow Security. IEEE Journal on Selected Areas in Communications, 21(1):5–19, 2003.
[10] L. Whitney. Microsoft Urges Laws to Boost Trust in the Cloud. 8301-1009 3-10437844-83.html.