Simultaneously Supporting Privacy and Auditing in Cloud Computing Systems

Tyrone Grandison
Proficiency Labs
Ashland, Oregon, USA
tgrandison@proficiencylabs.com

Sean Thorpe
School of Computing and Information Technology
University of Technology (UTech)
Kingston, Jamaica
sthorpe@utech.edu.jm

Leon Stenneth
Department of Computer Science
University of Illinois
Chicago, Illinois, USA
lstennet@cs.uic.edu

Abstract— **Over the last few years, cloud services have been steadily gaining traction in their use by commercial and non-commercial entities. As more and more sensitive or valuable processes, business functions and data move into the cloud, the need to improve threat identification and response, via auditing cloud transactions, increases. At the same time, the need for cloud users to protect the security and privacy of their resources has also intensified. In this paper, the problem of simultaneously supporting privacy and auditing in cloud systems is studied. Specifically, the paper discusses the guiding principles, fundamental concepts, and threat models for current cloud computing systems. Finally, we propose infrastructure that exploits a novel thin layer between the client and the cloud service provider to ensure that data storage, operation, and auditing does not reveal sensitive client information.**

Keywords:Auditing, Cloud Computing,Privacy,

I. Introduction

The dominance of cloud service providers and the presumption of cloud infrastructure as the de facto standard for firms over the last few years is common knowledge in Silicon Valley [1]. The economies of scale offered by cloud services enable small, agile teams to go from idea to product in a relatively short time [2]. However, this phenomenon has associated, and often unintended, consequences.

The first consequence is being slowly revealed to the general public with the multiple failures of Amazon Web Services [3, 4], which brought operations at Reddit, Quora, Github, Minecraft, Flipboard, Airbnb, Heroku, Netflix, Pinterest, FastCompany, FourSquare and a number of other companies to a screeching halt. The impact of these breakdowns range from mildly irritating to financially significant. For example, in a failure incident in 2011 [3], over 330 websites temporarily went down, while another outage in 2012 [4] had a substantial impact on over 100 companies, in terms of their brand, reputation and bottom line. In the long term, belief in guarantees of complete reliability in and total dependability on cloud systems is not realistic.

The second consequence is the increased need for security and privacy service level agreements (SLAs) from cloud service users. This arises from the fact that companies are archiving (and have archived) sensitive information and operations in the cloud. Tenants require legally binding agreements to ensure business continuity. Cloud outages indicate to cloud tenants that system errors (such as hardware and network episodes) and or external incidents (such as denial of service attacks) will occur and will most likely negatively impact their businesses. Cloud clients expect the number of security incidents on cloud systems to increase over time [5].

Some of the security and privacy SLAs that potential cloud service users discuss prior to service use include contractual minimums on reliability, rules around breach notification, specification of data isolation guarantees and assurances on the security controls and safeguards in place. We posit that the concerns around security and privacy represent the major impediments to the future of cloud systems.

In particular, we see the enablement of threat detection via audit log analysis and the preservation of the user's privacy (until such a time as when identification is warranted) as the initial primary issues that need to be addressed. This general goal of supporting both privacy and auditing has two major components. The first is how one facilitates the creation of privacy-preserving audit logs, which is necessary when the cloud user does not have full confidence in the cloud provider or their affiliated ecosystem. The second is enabling any random auditor to perform an audit in a privacy-preserving manner, which is needed when there is not complete trust in the auditor and the service provider. In this discourse, the authors seek to highlight both objectives.

The remainder of the paper is organized as follows: Section II describes guiding principles used when developing controls for cloud systems, Section III presents how cloud systems currently operate, Section IV discusses the general considerations when talking about privacy and auditing in clouds, Section V presents the architectural and design foundational concepts in the space, Section VI introduces our initial proposal, Section VII then highlights some issues that arise from our proposal, Section VIII discusses related work, and Section IX concludes the paper.

II. Guiding Principles

In accordance with the National Institute of Standards and Technology (NIST) Special Publication 800-144 [6] and the *Prime Directive*[1] of the cloud services industry [7], which states that *nothing interferes with the provision of services*, there are a set of well-defined principles that must be

[1] Each industry or sector has at least one axiom that must be adhered to by any system or subsystem, computerized or not, that is involved in the production of its main deliverable. This axiom is referred to as the *Prime Directive* for that industry.

followed when creating security, privacy and auditing controls or mechanisms for clouds, namely:

1) Seamless: The mechanism should integrate into the current mode of operation with minimal to no significant impact on the status quo.

2) Transparent: It should be clear to the cloud service user what the purpose of the mechanism is and when it is functioning.

*3) Elastic:*The mechanism should be able to scale to dynamically handle the request loads placed on the cloud service provider.

4) Low Impact: The inclusion of the mechanism should have a minor impact on the storage and performance of the cloud environment. This implies that the mechanism should be both lightweight, efficient in its operation and not significantly degrade the provider's or the cloud user's, performance.

5) Verifiable: An independent third party should prove the veracity of the actions of the mechanism. Thus, a trusted third party must be able to show that a timestamped, audit log record entry corresponds to a system activity or action that actually occurred in the specified time range.

The above list is not exhaustive and is grounded in cloud system use and operation, which will be better understood after the next section.

III. Cloud Service Operation

The first step in using a cloud service provider is acquiring the access credentials, which normally includes an access key ID, and a secret access key. For added security, some cloud providers provide the option for users to get session[2] and or federation[3] tokens to put time and permission constraints on the interaction with the provider's Application Programming Interface (API).

The cloud service user can then utilize his ID to make a call to the API of the cloud service provider. Typically, the user signs a request message, using their secret access key. The request message is the entity that contains the API call to the cloud service provider.

Currently, when metadata is generated (on the cloud user by the cloud provider), i.e. server audit logs are enabled, the identity of the user is normally not protected.

When cloud systems are audited, there is the need to include another party – the auditor – who may be internal, i.e. from the cloud user or the cloud provider, or may be an independent third party. The cloud auditor is tasked with examining cloud service controls with the intent to divine a legal opinion. These audits are performed with the intent to verify conformance to standards through the review of objective evidence [8]. In the case of an auditor, who may be employed by an enterprise that is a cloud tenant, these standards may include organizational policy and legislation specific to their industry and country. In the case of an external auditor, who may be a forensic investigator or country-specified compliance officer, these standards include the regulation relevant to their mission. In these logical domain audits, one may evaluate services provided by the cloud service provider in terms of security controls, privacy impact and performance.

IV. Design & Architectural Considerations When Enabling Auditing &Privacy

The design decisions that should be included when supporting and or building auditing and privacy (A&P) mechanisms in a cloud environment are:(1) the mechanism injection point (MIP), (2) the nature of the cloud service employed, (3) the transaction attack vector (TAV), and (4) the threat determination point (TDP).

A. The Mechanism Injection Point

The *mechanism injection point* refers to the location of the A&P controls. This is the location where enforcement of the auditing and privacy rules will be performed and the supplementary mechanisms, such as data structures are situated. The options for the MIP are: (1) at the cloud service user, (2) at the cloud service provider, or (3) at both locations.

Placing auditing and privacy mechanisms at the cloud user requires additional components to be either sent to the user from the provider or installed on the user's system as a prerequisite for using the cloud provider. For privacy controls in this configuration, further mechanisms should be in place to ensure atomicity and tamper-resistance of the control. Otherwise, one cannot guarantee: (1) that privacy enforcement always occurs (consistently), and (2) that the controls are not compromised and under the command of a malicious external party. Additionally, issues around the user's storage limits and log replication & recovery have to be addressed.

Having auditing and privacy functionality at the cloud provider requires that the same issues that arose in our prior discussion still need to be tackled. However, there is an additional need for attestation technologies that will enable the cloud user to affirm that the privacy and audit enforcement mechanisms are functioning correctly.

Experimental evaluation needs to be performed to determine the relative benefits and constraints around placement of the MIP. The authors can only conjecture that:(1) the "*at the provider*" placement would scale better in terms of infrastructure needed as the number of cloud users increases, and (2) the "*at the provider*" placement would be more resistant to system degradations as service use increases, which adheres to the principle of *Low Impact* for a longer period of time and in a larger number of use cases.

B. The Nature of the Cloud Service Employed

A cloud service user has a number of ways in which they can use the cloud resources. The current models being offered by providers include the delivery of Software-as-a-

[2] A session token is a token issued to a user calling the cloud service provider's API that allows them to issue calls to the API for a specified duration.

[3] A federation token is a token for a specified duration and permissions for a federated user or applications.

Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS).

SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. PaaS is a paradigm for delivering operating systems and associated services over the Internet without downloads or installation. IaaS involves outsourcing the equipment used to support operations, including storage, hardware, servers and networking components.

It should be noted that though these are the current set of delivery models, there are emerging models that interweave and extend the contemporary ones. For example, in the field of healthcare sensor networks, there is activity on Ecosystem-as-a-Service [9] offerings, where a community of related and relevant entities is presented to clients who need to bootstrap their environments. It should also be noted that there are many more emerging models [10], such as Monitoring as a Service (MaaS), Communication as a Service (CaaS), Desktop as a Service (DaaS), etc. In this paper, we will focus on the initial three NIST-sanctioned delivery models.

The impact of the model used on the cloud mechanisms being developed is that the mechanism type, and its' associated artifacts, vary from model to model. For SaaS users, A&P controls are built for the application level and operate on application objects. For PaaS users, A&P mechanisms are constructed for the operation system (OS) level and work on OS artifacts. For IaaS users, controls function at the network and hardware level and handle their associated objects.

C. *The Transaction Attack Vector*

The transaction attack vector refers to the class of transactions that are evaluated in the process of assessing a possible threat. There are two types of transaction attack vectors: *Requests* and *Consequences*.

Requests are calls made by the cloud user to utilize the cloud service. *Consequences* are the results of the *Requests* made by the cloud user, i.e. the results returned from the cloud service provider.

Enforcing an auditing or privacy control on *Requests* is done when there is a concern around the state of the cloud user. This activity will help to determine if the user has been compromised or not. Examining *Consequences* helps to ascertain if the cloud service provider has been or is under attack. A holistic approach is always recommended. Hence, a system that addresses both *Requests* and *Consequences* is the expected norm. However, for a myriad of reasons, it is understandable that one may be chosen over the other, depending on the specific needs, interactions and constraints of the user and provider.

D. *The Threat Determination Point*

The threat determination point refers to the location where the analysis of the recorded privacy and audit events occurs, i.e. the location where breach detection and notification happens. Note that the TDP is different from the MIP; where the MIP is the location at which the privacy enforcement (and logging) occurs and auditing is enabled (so that logs can be created), the TDP is the location where the logs are examined, (by a third party, i.e. the auditor) to ascertain if a suspicious activity has taken place, and where privacy compliance is determined.

For the purposes of this discussion, the TDP may be either at the cloud service user, at the cloud service provider, at the auditor or distributed between the user and provider. As in the MIP case, having the TDP at the cloud user necessitates tamper-resistance of the threat determination mechanisms. Placement of the TDP at the provider or at an external auditor calls for the associated controls to be able to provide proof or evidence to the affected parties of a breach.

As with the MIP, further research needs to be done to ascertain the relative advantages and disadvantages of one's placement of the TDP.

Now, we are poised to introduce the fundamental concepts and constructs that influence our proposal.

V. Fundamental Concepts & Constructs for Privacy & Auditing In Cloud Systems

Privacy is "the claim of individuals, groups, or institutions to determine for themselves when, and to what extent, information about them is communicated to others" [11]. Thus, at the core of the concept of privacy is the exercising of control over the disclosure and use of data; such that these items are protected from unintended eyes and from being used for unsanctioned purposes.

Auditing is the systematic process of objectively obtaining and evaluating evidence regarding assertions about actions and events to ascertain the degree of correspondence between those assertions and established fact and communicating the results to interested users. At the core, auditing is about (metadata) collection (into audit logs), data extraction (of said logs), data analysis in the context of some standard (e.g. law or interesting activity, such as a security breach), and results generation and dissemination.

As previously mentioned, the process of creating a system that supports both auditing and privacy involves 1) the creation of evidence, i.e. audit log files, in a privacy-preserving manner (hereafter referred to as *Task 1*) and, 2) the enablement of a privacy-preserving process for conducting audits (hereafter referred to as *Task 2*). In order to accomplish both tasks, we need to understand the current strategies for doing both.

A. *Task 1 - The Current (Privacy)Strategy*

In the standard cloud environment, the data items that often need to be protected may fall into one of three arbitrary categories: identity data, location data, and confidential data [14]. The basic conceptual techniques for ensuring data privacy involve either 1) constructing and deploying a methodology that allows an individual's data to be hidden in a much larger crowd of larger data [12, 13], or 2) building and delivering a solution that securely and directly transforms individual data items to their alternate representations or surrogates [14]. After this transformation process, the data is normally stored (in the audit log).

B. *Task 2 - The Current (Auditing)Strategy*

Though auditing of cloud systems is relatively new and there are no established standards, techniques are being adapted from typical IT audit processes; and digital forensics investigations are being applied to cloud computing [8, 10]. The processes are informed by input from NIST, the Information Systems Audit and Control Association (ISACA), the Cloud Security Alliance (CSA), the Federal Risk and Authorization Management Program (FedRamp) from America's General Services Administration (GSA), and the European Network & Information Security Agency (ENISA).

The emerging approach to streamline the cloud audit process includes the definition of a key set of terms for service level agreements (SLAs) that enable the cloud auditor to enhance the examination and verification capabilities [15]. In order to understand the SLAs, the relationship between the cloud consumer and the cloud service provider (sometimes seen as the cloud carrier based on the jurisdiction) should be evaluated. The cloud carrier acts as that intermediary that provides connectivity and transport of cloud services between the cloud customers and providers.

Typically, the cloud provider arranges two unique SLAs, one with a cloud carrier (e.g. SLA2) and the other with a cloud requester/consumer (e.g. SLA1). A cloud provider request dedicated and encrypted connections to ensure the cloud services are consumed at a consistent level according to the contractual obligations with the cloud consumers. In this case, the provider may specify its requirements on capability, security, privacy, flexibility and functionality in SLA2 in order to provide essential requirements in SLA1.

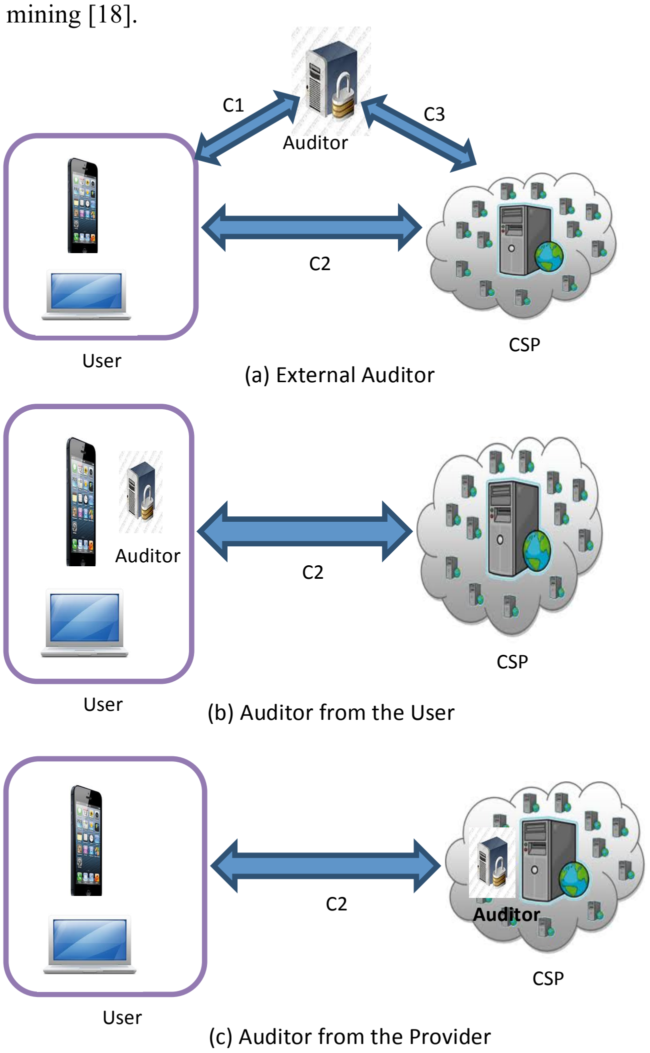
In a traditional case, carriers are not likely to be involved with the cloud audit investigation. However, they can play a useful role in providing pre-investigative and supportive capabilities, such as evidence transport, claim of custody, and inter-cloud forensic capabilities.

The current ways that both tasks are performed directly influence the threat model.

C. *The Current Threat Models*

In the general threat model, there are four logical entities taking part: 1) the cloud user (otherwise called the cloud consumer or the cloud tenant), 2) the cloud service provider, 3) the auditor, and 4) the adversary. We use the term logical because there may be instances where 1) the adversary may be one of the other three parties (or an external agent), or 2) the auditor may be an agent or representative of the cloud user, the cloud service provider (i.e. when either party may be performing their own internal audit actions) or an external entity.

Our base assumption is that, in the typical case, the parties involved have no (or very little) trust in each other. However, we recognize that in some cases it may be more expedient to trust the auditor if they are an external third party; adopting as a stance of "trust, but verify" as often as possible. The attack or threat model in this context (Where both privacy and auditing co-exist) is different from location privacy preservation [16, 17] and privacy preserving data mining [18].



(a) External Auditor

(b) Auditor from the User

(c) Auditor from the Provider

Figure 1.Threat Model Configurations

In location privacy, the traveler's location or the query's location is the quasi-identifier[4] and in this environment, background information about the user is the quasi identifier [18]. In our situation, the notion of a quasi-identifier in the threat model is contextual (i.e. depending on the cloud system and the users, the set of quasi-identifier varies). Likewise, the adversaries and their intentions are different. In location privacy and privacy preserving data mining the assumption is that adversary is external to the system and can combine background information (e.g. census data) with partially anonymized data to positive identify an individual [18]. While in this context, the assumption is that the adversary may be internal or external to the system. In one example, the cloud provider has been compromised, and is functioning as an adversary, and deliberately deletes rarely accessed files that belong to other cloud users. In another

[4] A quasi-identifier is an attribute or a combination of attributes within a dataset that are not structural unique, i.e. not primary keys, but might be empirically unique and therefore in principle uniquely identify a user.

example, an external adversary loiters between the communication channel of the user and the cloud service provider and can listen to *request* and *consequence* transactions.

We consider a cloud service that includes four entities: (1) cloud service provider (CSP), (2) cloud user, (3) auditor, and adversary (see Fig. 1). There can be several communication channels in the model. These channels are represented as C1, C2, and C3 (Fig. 1a).

User - The cloud user has large amount of data to be archived in the cloud or processing activities to be outsourced to the cloud.

CSP – The cloud service provider, for example Amazon Web Services or Eucalyptus [19], provides storage space and other services and has significant computation resources.

Auditor– Auditors are experts beyond the ordinary cloud users and may access the user-related cloud resources on behalf of the users, when requested.

C1, C2, and C3 – In the model, there is communication between the entities involved. For example, when the user has cloud storage or processing requirements, C2 is activated. Responses from the CSP back to the user are also on this C2 channel. For the external auditor scenario, Fig. 1(a), an auditing request from the user to the auditor is sent via channel C1, the auditing procedure occurs via C3. In situations, where the auditor is from the user or CSP, i.e. figures 1(b) and 1(c), auditing requests and the audit itself occurs on the C2 channel.

Adversary – Though not shown in Fig. 1, the adversary's intention is to decipher, delete, modify, or access the cloud user's data and or to divine insight from the user's processing needs and operations.

Let us discuss how the model works. Users rely on the CSP's for cloud services. Additionally, they may dynamically send updates to existing data that they own in the cloud. Since this data is sent over a channel (i.e. C2) a number of privacy breaches are possible. As explained, adversaries can attack the API calls while the *requests* or *consequences* are traversing C2. This is true for all the threat configurations discussed earlier.

After the data arrives in the cloud, since it is assumed that the data is now out of the user's jurisdiction, it is challenging for the user to validate the correctness of its offshore data, unless the CSP provides trustworthy mechanisms to do so. As a failsafe, users may recourse to the auditor to ensure the security and correctness of their outsourced data is maintained. Realistically, the auditor and its associated channels can be a region of adversarial attack. For example, adversaries may loiter on the channel between the auditor and the CSP (see Fig. 1a). For this case, the auditor is assumed to be reliable and independent of the cloud user and the CSP. They (i.e. cloud auditor) audit the data in the CSP on the users' behalf with constraint on the amount of data it (i.e. auditor) accesses.

In most cases, this CSP adheres to the correct data archiving protocols and standards [20]. However, rarely they may deliberately put their own benefits above others and say delete some files that are not accessed in a long time or fail to report that they (i.e. CSP) lost the users' files during random hardware failures. Furthermore, in order to maintain reputation, the CSP may hide data corruptions caused by hacking. Thus, considering a reliable and independent third party model to audit the CSP on behalf of the user adds a new dimension to privacy preservation auditing in cloud systems.

In other cases, the auditor can be an agent of the user (Fig. 1b) or an agent of the CSP (Fig. 1c). Consequently, the threat model changes since for example, in the case of Fig. 1c, the auditor can collude with the CSP. The case where the auditor is an agent of the user (Fig. 1c) is more realistic than Fig. 1b since the auditing task is normally a function for the user that is supported by the CSP.

VI. Proposed Approach

In the typical scenario, we assume that there are three principal distinct entities, as shown in Fig. 1(a): the auditor, the cloud user, and the cloud service provider (CSP). Additionally, we assume the introduction of a Public Key Infrastructure (PKI), whether a trusted third party (our preference) or one provided by the CSP.

In order to accomplish *Task 1* and to incorporate the guiding principles identified in Section II, we propose adding a thin layer in both the client and the CSP that ensures that the CSP stores data and operates on them without knowing sensitive information about the client. This thin layer on the CSP intercepts all requests coming into the CSP and applies privacy-preserving functions to the request before sending a transformed (and privacy-compliant) version of the API call to the native API, i.e. the original API. In order to enable this privacy-preservation, a simple routine could apply a hash function on the user-supplied ID [14], and then utilize that as a base to query the PKI for the user's public key, which is then used to encrypt all sensitive information in the request (Fig. 2). The steps that represent the process followed by the user and the CSP are outlined in Table I.

The privacy-preserving API would be installed on the client securely stores the user's secret key and decrypt the encrypted responses sent from the CSP (Fig. 2). We assume that the channel C2 is encrypted, as well as communications between the PKI and both the CSP and user (with standard, secure web protocols), in order to reduce eavesdropping on the wire. As each user's data is encrypted with their own public key, not only does it prevent the CSP from seeing sensitive data, it also prevents other cloud tenants from seeing the user's data. We recognize that operations on encrypted data currently have performance and other limits. However, we presume that measures can be employed on the CSP to mitigate or minimize these issues.

For *Task 2*, the TDP is external to both the user and the CSP. We make the complicit assumption that the auditor operates within an independent physical environment where access control, trust and threat determination in this domain is presumably well understood as a part of enabling the audit provision in the first place.

In order to ensure that the auditor is unaware of both the location and identity of the user requesting an audit, we suggest the use of network traffic analysis preventative tools such as Tor [21] and the use of pseudonyms.

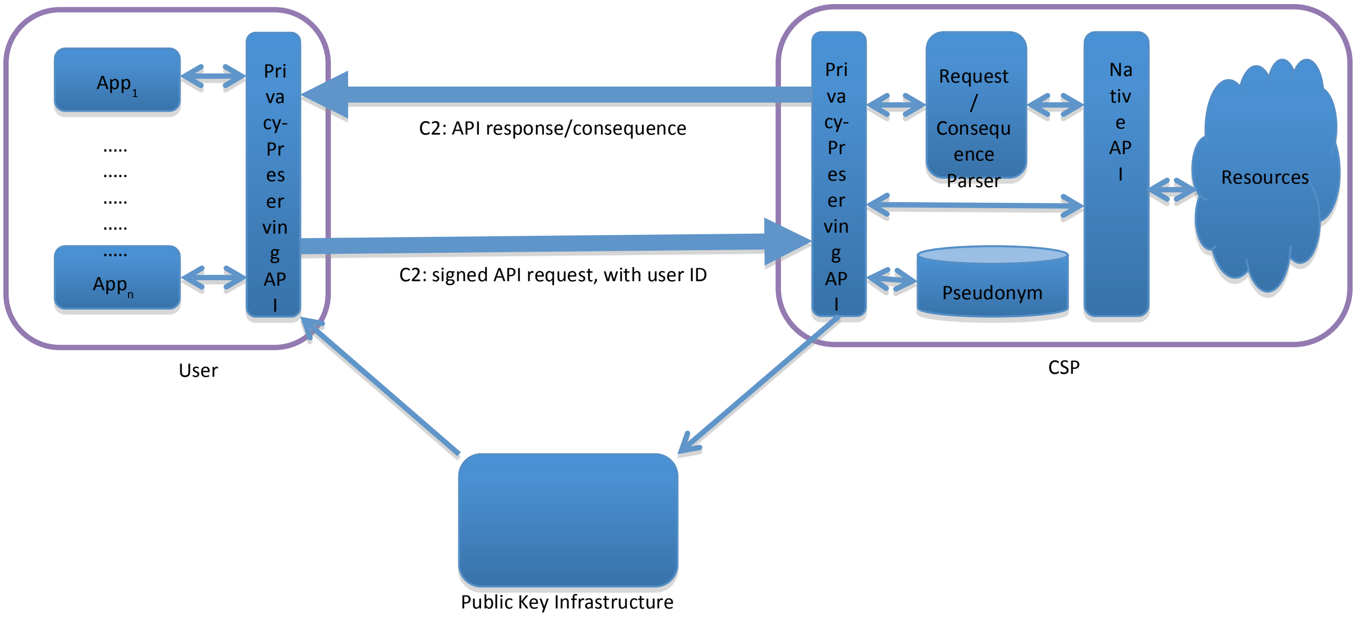

App1
.....
.....
.....
.....
.....
Appn
Privacy-Preserving API
C2: API response/consequence
C2: signed API request, with user ID
User
Privacy-Preserving API
Request / Consequence Parser
Pseudonym
Native API
Resources
CSP
Public Key Infrastructure

Figure 2.Proposal for *Task 1* – Generation of Privacy-Preserving Logs

TABLE I. PSEUDOCODE FOR TASK 1

```
1. User: Sends a signed request with ID to the CSP – ($request_A$, $ID_A$, $signature_A$)
2. CSP: If verify($signature_A$) is false, return ERR to User A
3. CSP: If verify($signature_A$) is true
   1. If ID is in pseudonym_table
      1. Set $P_A$ to User A's pseudonym
      2. Query PKI for A's public key using $P_A$
      3. Set A's public encryption key, $key_A$, to the key returned from the PKI
   2. If ID is not pseudonym_table
      1. Generate a pseudonym $P_A$ and store it in pseudonym_table
      2. Use $P_A$ to generate public key pair for A
      3. Have the PKI send A's secret key, $SK_A$, to the user A
      4. Set $key_A$, to A's public key
   3. Parse $request_A$ to extract an array of sensitive items, $S(R_A)$
   4. Encrypt $S(R_A)$ using $key_A$
   5. Rewrite $request_A$ using encrypted $S(R_A)$ parameters
   6. Send rewritten $request_A$ to the former API
   7. Receive response, $CS_A$, from former API
   8. Send $CS_A$ to User A
4. User: Parse $CS_A$ to extract array of sensitive items, $S(CS_A)$
5. User: Decrypt $CS_A$ using $SK_A$
6. User: Rewrite $CS_A$ and send to client
```

TABLE II. PSEUDOCODE FOR TASK 2

```
1. User: Sends a signed request with ID to the CSP for $P_A$
2. User: Sends a request to the auditor using $P_A$
3. Auditor:  Sends request for logs related to $P_A$
4. CSP: Queries PKI to acquire A's public key, $key_A$
5. CSP: Transforms audit request by encrypting sensitive parameters
6. CSP: Retrieves logs related to $P_A$ and sends it to Auditor
7. Auditor: Analyzes logs with encrypted sensitive data
8. Auditor: Disseminates results
```

The steps that the entities must take in enabling the audit process in a privacy-preserving manner (*Task 2*) are specified in Table II. As with *Task 1*, we assume that the communications between all parties is occur over secure channels. We observe that the logs to be analyzed by the auditor are directly correlated to the nature of the cloud service being employed the user. For example, if the user is leveraging a PaaS model, then the operating system logs are the targets of analysis. Similarly, for SaaS model users, logs of the activities of application objects are analyzed. Comparable logic should be applied to the case of IaaS model users.

We recognize that more research needs to be done in order to enable sophisticated auditing of encrypted, sensitive data, by auditors. We also recognize that creating specialized operations for specific data types and for specific analytic purposes is possible and has been an investigative space with great promise [22]. As this is just one proposed architectural configuration, we acknowledge that there are other possible proposals that may include the auditor having direct communication with the PKI.

Having presented a proposal for simultaneously supporting both privacy and auditing in cloud computing systems, we now offer further discussion on potentially interesting resultant concerns.

VII. DISCUSSION

The issue of how much data the user should reveal to the auditor for auditing to be possible is a research issue. The decisions taken in our proposal seek to provide a level of privacy that is expected by users when interacting with social networking clouds and general online companies [23, 24]. However, for commercial and other reasons, user expectations may not translate into realized user protections. We recognize that non-data encryption strategies, such as public-key homomorphic authentication [25-28], which is the "state of the art" and reveals zero information to the auditor, may be employed in this environment. Other techniques, such as the skip lists [29], also reveal a small amount of the data to the auditor. Generally, there tends to be a tradeoff between the amount of data revealed, computational cost, success rate, and auditing time. We believe that other dynamic strategies such as *k*-anonymity [12] with diversity constraints that reveal some information to the auditor may produce interesting results. Optimizations to all these techniques also offer an interesting sphere for research.

In a special case where the CSP provides functionality for spatial and temporal data, cloaking techniques may be considered. For example, cloud users that handle spatial and temporal data and utilize the CSP may require customized ways to audit their spatial and temporal data. The cloud user may transit a region enclosing all the location points to be audited to the auditor. Since the user transmits a region instead of location points to the auditor, then the user's precise information is not revealed to the auditor. Given the region, the auditor may then generate $k \in \mathbb{N}$ location points, which follow a random distribution, within the region. The auditor may then perform supported computations, such as average Euclidian distance and standard deviation, between the *k* location points and the original points archived by the CSP. The auditing results and the *k* randomly distributed location points may then be sent to the user for further validation. Observe that, (1) the auditor did not receive a copy of the user's data and (2) a copy of the user's data was not transmitted over the channel during the auditing process. Overheads may include the generation of randomly distributed location points by the auditor including validation by the cloud user.

The first step in conducting an audit (and in collecting evidence for a forensics investigation) is the process of deciphering and understanding the interactions between the cloud entities. These cloud actor interactions and the linear dependencies between them provide an indicative trail of potential evidence that can be collected, as suggested by Liu et al. [8]. The interaction scenarios are detailed views of the various cloud organizational dimensions described by Ruan et al. [15] and are analyzed within the context of SLAs, the guidance of internal and external investigational procedure, and the forensic artifacts. Analysis of these interactions, based on fuzzy logs, is an area that requires further work.

Our proposal sought to strike a balance between the guiding principles presented in Section II. However, we cannot at this point definitely state that these principles have been optimized. The introduction of a single layer within the client and auditor to abstract away the details that enable the creation of private logs may or may not meet the principles of seamlessness and transparency for some cloud users. Thus, studies to determine the level of tolerance that users have for various techniques is an area of open research. The principles of elasticity and low impact have to be tested on CSP of varying dimensions and for user populations scaling up to realistic sizes. Though there has been significant work on the formal proof of the verification of logs [30], applying them to the cloud computing environment is an area with its own unique set of concerns that need to be factored into the discourse and handled as an independent paper.

VIII. Related Work

In [31], the authors consider encryption-based homomorphic authenticators for auditing outsourced data and suggest randomly sampling a few blocks of the file (*Task 2*). However, the public auditability in their scheme demands the linear combination of sampled blocks exposed to the external auditor. When used directly, their protocol is not provably privacy preserving, and thus may leak user data information to the auditor.

Wang et al. [32] propose to combine the Boneh–Lynn–Shacham (BLS) based homomorphic authenticator with the Merkle Hash Tree (MHT) to support both public auditability and full data dynamics (*Task 2*). Further, Erway et al. [27] developed a skip lists based scheme to enable provable data possession with fully dynamics support. However, all their protocol requires the linear combination of sampled blocks just as [26], and thus does not support privacy-preserving auditing on user's outsourced data.

The works in [26, 28] considered the trusted third party framework and introduces a clever technique independent to data encryption. Their public key homomorphic authenticator enables the auditor to perform auditing without requiring a local copy of the data. Since the data is not required, it drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches.

In some auditing approaches, verification is supported using ring signatures [28] across shared groups. If the groups are situated within an on-site private cloud domain, the verification process is manageable. However when we consider off-site private and public clouds, as well as the hybrid of these deployments this becomes a complex issue. The complexity of the issue is grounded in the fact that verification processes have scaled [29]. This scaling affects the ability to maintain a privacy preserving cloud ecosystem. Against this background, the auditing of these logical domains requires that verification will have to be modeled independently across the IaaS, PaaS and SaaS layers of the data cloud stack as a useful measure. It also importantly means that for such verification controls to be enforced, a scalable location privacy preservation model across all the stack layers need to be well defined. Where this paper provides the general considerations of such a model, further evaluations are covered in independent work.

Location privacy and privacy preservation data mining have been addressed in our prior works [23-25] and provides suitable terms of reference for incorporating those concerns for cloud audit logs. There are cases where location privacy preserving schemes can be adopted for privacy preserving cloud auditing. In the trusted third party privacy preserving location based systems [23, 25], the user communicates precise location to the trusted third party who anonymizes the location data before submission to the location based server. Consequently, location adversaries that loiter between the trusted third party and the location-based server have challenges to decipher the anonymized location data we believe these concerns are no less trivial for the cloud logging environments and wholeheartedly adapt these considerations. Anonymizing the location data involves spatial and temporal cloaking techniques and location based *k*-anonymity with diversity considerations [23, 24] and may present useful approach for handling the audit cloud log clouds particularly as a forensic concern.

IX. Conclusion

As cloud services become more prevalent in the IT landscape, so does the need for enabling privacy and forensic auditing in these logical abstract domains. In this paper, we provide guidelines to help in building these controls so as to ensure their sustained use and relevance in these ubiquitous environments. We highlight the current state of affairs in the relevant spaces, set the foundational concepts and constructs and present a proposed approach to the problem. This preliminary work in the space portends interesting future research directions and we hope it stimulates ideas and collaborations.

References

[1] D. Linthicum. "The cloud computing revolution will not be televised". InfoWorld. August 19th, 2009. Retrieved from https://www.infoworld.com/d/cloud-computing/cloud-computing-revolution-will-not-be-televised-824 on May 19th, 2013.

[2] D. Linthicum. "Cloud computing: The semi-secret economic equalizer". InfoWorld. November 20th, 2011. Retrieved from https://www.infoworld.com/d/cloud-computing/cloud-computing-the-semi-secret-economic-equalizer-207407 on May 19th, 2013.

[3] R. Liu. “Amazon Cloud Failure Going on Day Two”, SlashGear, April 22nd, 2011. Retrieved fromhttp://www.slashgear.com/amazon-cloud-failure-going-on-day-two-22147919/onMay 19th, 2013.

[4] R. MacMillan. “Amazon Cloud Goes Down Again, Breaks Foursquare and Others”, Wired Magazine, October 22nd, 2012. Retrieved from http://www.wired.com/wiredenterprise/2012/10/amazon-web-services/ on May 19th, 2013.

[5] D. Linthicum. "As cloud use grows, so will rate of DDoS attacks". InfoWorld. February 5th, 2013. Retrieved from http://www.infoworld.com/d/cloud-computing/cloud-use-grows-so-will-rate-of-ddos-attacks-211876 on May 19th, 2013.

[6] W. Jansen and T. Grance. “Guidelines on Security and Privacy in Public Cloud Computing”, National Institute of Standards and Technology (NIST), Special Publication 800-144. December 2011. Retrived from http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf on May 19th, 2013.

[7] T. Grandison and J. Davis. "The Impact of Industry Constraints on Model-Driven Data Disclosure Controls". The Proceedings of the 1st International Workshop on Model-Based Trustworthy Health Information Systems (MOTHIS) 2007, Tennessee. Sept 2007.

[8] F. Liu, J.Tong, J.Mao, J.Bohn, R.Messina, J.Badger and D.Leaf. “NIST Cloud Computing Reference Architecture”. National Institute of Standards and Technology, Special Publications No. 500-291.

[9] T. Grandison, P. S. Hsueh, L. Zeng, H. Chang, H. Chen, C. Lan, H. Pai and L. Tseng. "Privacy Protection Issues for Healthcare Wellness Clouds" in Privacy Protection Measures and Technologies in Business Organizations: Aspects and Standards. Editor: George Yee. 2011.

[10] B. Halpert. "Auditing Cloud Computing: A Security and Privacy Guide". Vol 2. John Wiley & Sons, 2011.

[11] A. Westin, . "Privacy and Freedom”, New York: Atheneum, 1967.

[12] P. Samarati and L. Sweeney. “Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression”. Technical report, SRI International, 1998.

[13] I. Goldberg. “Privacy-enhancing technologies for the Internet, II: Five years later”. In Privacy Enhancing Technologies (pp. 1-12), 2003.

[14] A. Lysyanskaya, R. L. Rivest, A. Sahai and S. Wolf. “Pseudonym systems”. In Selected Areas in Cryptography (pp. 184-199). Springer Berlin Heidelberg, 2000.

[15] K.Ruan, J.Carthy and T.Kechadi. “Cloud Forensics: Key terms for Service Level Agreements” Advances in Digital Forensics VIII,Springer.

[16] L. Stenneth, P. S. Yu and O. Wolfson. MobiPriv: Mobile systems location privacy: “MobiPriv” a robust k anonymous system. 6th IEEE International Conference on Wireless and Mobile Computing, Networks and Communication (WiMob), 2010.

[17] L. Stenneth and P. S. Yu. Mobile Systems Privacy. "MobiPriv" A Robust K-Anonymous System For snapsot and continious querying location based systems. Transaction on Data Privacy, volume 5, issue 1, 2012.

[18] C. Aggarwal and P. S. Yu. “Privacy-preserving data mining:models and algorithms”. Kluer academic publishers, 2010.

[19] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and Dmitrii Zagorodnov. "The eucalyptus open-source cloud-computing system." 9th IEEE/ACM CCGRID 2009.

[20] D. D. Crouse, H. G. Coverston and J. M. Cychosz. "Archiving file system for data servers in a distributed network environment." U.S. Patent No. 5,764,972. 9 Jun. 1998.

[21] R. Dingledine, N. Mathewson, and P. Syverson. “Tor: The second-generation onion router”. Naval Research Lab Washington DC, 2004. Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA465464on May 19th, 2013.

[22] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. “Order preserving encryption for numeric data”. in Proc. of the 2004 ACM SIGMOD international conference on Management of data (pp. 563-574).

[23] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. “Analyzing Facebook privacy settings: User expectations vs. reality”. The Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement, pp. 61-70.

[24] T. Grandison, and R. Bhatti. “HIPAA Compliance and Patient Privacy Protection”. Studies In Health Technology And Informatics, 160 (Pt 2), 884-888.

[25] H. Shacham and B. Waters. “Compact proofs of retrievability,” in Proc. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107.

[26] C. Wang, Q. Wang, K. Ren, and W. Lou. “Privacy preserving Public Auditing for Data storage security in Cloud Computing”. IEEE INFOCOM 2010.

[27] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia. “Dynamic provable data possession,” in Proc. of CCS, 2009.

[28] B. Wang, B. Li and H. Li. “Oruta.: Privacy Preserving Public Auditing for shared data in the Cloud”. Technical Report, University of Toronto, 2011.

[29] D. Boneh, C. Gentry, B. Lynn and H. Schacham. “Aggregate and Verifiably Encrypted Signatures from bilinear Maps”. The Proceedings of EUROCRYPT, Springer Verlag (2003), pp. 416-432.

[30] H. Barringer, A. Groce, K. Havelund and M. Smith. “Formal analysis of log files”. Journal of aerospace computing, information, and communication,7(11), pp. 365-390, 2010.

[31] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. “Provable datapossession at untrusted stores,” Cryptology ePrint Archive, Report 2007/202, 2007, http://eprint.iacr.org/

[32] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou. “Enabling public verifiability and data dynamics for storage security in cloud computing,” in Proc. of ESORICS’09, Saint Malo, France, Sep. 2009.