2010 29th IEEE International Symposium on Reliable Distributed Systems

Protection of Identity Information in Cloud Computing without Trusted Third Party

Rohit Ranchal, Bharat Bhargava
Department of Computer Science
Purdue University
West Lafayette, IN, USA
{rranchal, bbshail}@purdue.edu

Lotfi Ben Othmane, Leszek Lilien
Department of Computer Science
Western Michigan University
Kalamazoo, MI, USA
{lotfi.benothmane, leszek.lilien}@wmich.edu

Anya Kim, Myong Kang
Naval Research Laboratory
Washington, DC, USA
{anya.kim, myong.kang}@nrl.navy.mil

Mark Linderman
Air Force Research Laboratory
Rome, NY, USA
mark.linderman@rl.af.mil

Abstract—**Cloud computing allows the use of Internet-based services to support business processes and rental of IT-services on a utility-like basis. It offers a concentration of resources but also poses risks for data privacy. A single breach can cause significant loss. The heterogeneity of "users" represents a danger of multiple, collaborative threats.**

In cloud computing, entities may have multiple accounts associated with a single or multiple service providers (SPs). Sharing sensitive identity information (that is, Personally Identifiable information or PII) along with associated attributes of the same entity across services can lead to mapping of the identities to the entity, tantamount to privacy loss.

Identity management (IDM) is one of the core components in cloud privacy and security and can help alleviate some of the problems associated with cloud computing. Available solutions use trusted third party (TTP) in identifying entities to SPs. The solution providers do not recommend the usage of their solutions on untrusted hosts.

We propose an approach for IDM, which is independent of TTP and has the ability to use identity data on untrusted hosts. The approach is based on the use of predicates over encrypted data and multi-party computing for negotiating a use of a cloud service. It uses active bundle—which is a middleware agent that includes PII data, privacy policies, a virtual machine that enforces the policies, and has a set of protection mechanisms to protect itself. An active bundle interacts on behalf of a user to authenticate to cloud services using user's privacy policies.

Keywords- active bundle; computing predicates; cloud computing; identity management system; multi-party computing; privacy; security.

I. INTRODUCTION

A. Privacy in Cloud Computing

The growing popularity, continuing development and maturation of cloud computing services is an undeniable reality. Information stored locally on a computer can be stored in the cloud, including word processing documents, spreadsheets, presentations, audio, photos, videos, records, financial information, appointment calendars, etc.

A cloud *service provider* (*SP*) is a third party that maintains information about, or on behalf of, another entity. Trusting a third party requires taking the risk of assuming that the trusted third party will act as it is expected (which may not be true all the time). Whenever some entity stores or processes information in the cloud, privacy or confidentiality questions may arise [2].

Privacy in cloud computing can be defined as "the ability of an entity to control what information it reveals about itself to the cloud (or to the cloud SP), and the ability to control who can access that information"[3].

Numerous existing privacy laws impose the standards for the collection, maintenance, use, and disclosure of *personally identifiable information* (*PII*) *th*at must be satisfied even by cloud SPs [2]. (PII is commonly known as *identity information*.) Due to the nature of cloud computing, there is little or no information available in a cloud to point out where data are stored, how secure they are, who has access to them, or if they are transferred to another host (if that host can be trusted).

A cloud cannot be used for storing and processing data and applications if it is unsecure. The major problem regarding privacy in cloud is how to secure PII from being used by unauthorized users, how to prevent attacks against privacy (such as identity theft) even when a cloud SP cannot be trusted, and how to maintain control over the disclosure of private information.

Handing sensitive data to another company is a serious concern. Are data held somewhere in the cloud as secure as data protected in user-controlled computers and networks? Cloud computing can increase the risks of security breaches. Knowing who has user's personal data, how they are being accessed, and the ability to maintain control over them prevents privacy breaches of PII, and can minimize the risk of identity theft and fraud [3].

We provide more details about privacy in cloud computing in [3]. We achieve a solution to the privacy problem that we investigate in this paper in [3] through the use of an entity-centric approach. The approach that we propose in [3] is based on anonymous identification, which is used to mediate interactions between the user and cloud services. The approach uses active bundle [16], which

1060-9857/10 $26.00 © 2010 IEEE
DOI 10.1109/SRDS.2010.57

enforces the policies and uses a set of protection mechanisms to protect the sensitive data.

B. Identity Management in Cloud Computing

A cloud user has to provide sensitive personal information (e.g., name, home address, credit card number, phone number, driver's license number, date of birth, etc.) while requesting services from the cloud. This leaves a trail of PII that can be used to uniquely identify, contact, or locate a particular user, which—if not properly protected—may be exploited and abused [2].

IDM is the key to cloud privacy and security but IDM in cloud is more complex than in traditional web-based systems since the users hold multiple accounts with different SPs or with a single SP. The traditional model of application-centric access control, where each application keeps track of its collection of users and manages them, is not acceptable in cloud-based architectures; sharing PIIs of the same entity across services can lead to mapping of PIIs to the entity, tantamount to a privacy loss [4].

C. Motivating Scenario for Identity Management in Cloud Computing

To use a cloud service, a user needs to authenticate herself to it. The user has to give away some private information, which uniquely identifies the user to SP, the other party. This is user's PII. Obtaining the user's PII gives some assurance to SPs about the users' identity, which helps SP to decide whether to permit access to its service or not.

Since identity information is a key for opening access to resources and for paying for them, it can lead to serious crimes involving identity theft [2] if misused or compromised. The purpose of an IDM system is to decide upon the disclosure of this information in a secure manner.

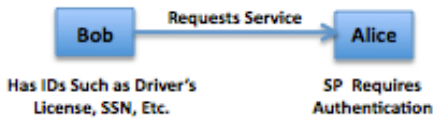

Figure 1. User-SP Interaction

Fig. 1 shows an example of authentication that uses PII. In the example, Bob wants to use a service. Bob has to disclose some of his PII, which uniquely identifies him to the SP. However, he doesn't want to disclose all his PII. The main problem for Bob is to decide which portion of his PII should he disclose, and how to disclose it in a secure way.

D. Related Work

We looked into the most known solutions for IDM. The detailed discussion is present in [3].

1) *PRIME:* Privacy and Identity Management for Europe (PRIME) [7] provides privacy-preserving authentication using a TTP, named IdP.
2) *Windows CardSpace:* Windows CardSpace [11] treats every digital identity as a security token, which consists of a set of claims (such as a username, a full name, address, SSN, etc). The tokens prove to SP that the claims belong to the user presenting them.
3) *Open ID:* Open ID [5, 6] is a decentralized authentication protocol that helps cloud users in managing their multiple digital identities with a greater control over sharing their PIIs. A user has to remember one username and password—an OpenID—and can log onto websites with this OpenID.

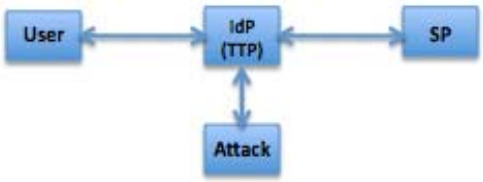

Figure 2. Architecture of and Attacks on Existing IDM Systems

Different solutions use different ways of sending user's PII for negotiation with the SPs. The common ways are:

1) *Use of a Trusted Third Party (TTP).* Most of the solutions that we studied use TTP for verifying or approving PII. The major issues with such approach in cloud computing (shown in Fig. 2) are: (a) TTP could be a cloud service, so SP could be TTP; therefore, TTP may not be an *independent* trusted entity anymore; (b) using a single TTP is a centralized approach with its inherent danger that compromising TTP results in compromising all PIIs of its users as well.
2) *Prohibiting untrusted hosts.* A client application holding PII must be executed on a trusted host to prevent malicious hosts from accessing PII.

E. Selected Research Problems

The research problems that we address in this paper are:

1) *Authenticating without disclosing PII:* When a user sends PII to authenticate for a service, the user may encrypt it. However, PII is decrypted before an SP uses it. As soon as PII is decrypted, it becomes prone to attacks.
2) *Using services on untrusted hosts:* The available IDM solutions require user to execute IDM from a trusted host. They do not recommend using IDM on untrusted hosts, such as public hosts. Since in cloud computing data may reside anywhere in the cloud (on any host), this issue needs to be addressed. E.g. User herself may be on a cloud Virtual Machine.

Note that goal in the paper is to assure that IDM does not use TTP for verifying credentials. This implies that IDM could use TTPs for other purposes, such as the use of a TTP by IDM for management of decryption keys.

In this paper we address some of the similar selected research problems as in [3]. However, we provide a different approach for addressing them.

F. Contribution and Paper Organization

We propose an approach for IDM in cloud computing that: (a) does not require TTPs; (b) can be used for on untrusted or unknown hosts; and (c) uses encrypted data when negotiating the use of PII for authentication to services in cloud computing.

The paper is organized as follows: Section II describes the proposed approach for protecting PII in cloud computing. Section III discusses the advantages of the proposed approach. Section IV shows the resilience of the proposed approach to two attacks. Section V concludes the paper.

II. PROPOSED IDM APPROACH FOR PROTECTING PII IN CLOUD COMPUTING

This section describes the proposed IDM approach, which is based on IDM using the *active bundle* scheme, computing predicates over encrypted data and multi-party computing.

The salient features of the approach are:

1) *Ability to authenticate without disclosing unencrypted data.* This is achieved by using predicate over encrypted data.
2) *Ability to use identity data on untrusted hosts.* This is achieved through the use of the active bundle scheme. An active bundle has a self-integrity check mechanism, which triggers apoptosis (a complete self-destruction) or evaporation (a partial self-destruction) when the check fails.
3) *Independence of TTPs.* This is achieved through the use of multi-party computing, in which secrets are split into shares distributed to different hosts.

A. Use of Predicates with Encrypted Data and Multiparty Computing

We use a predicate encryption scheme and multi-party computing for giving answer to predicates about PII.

Shamir [14] proposes threshold secret sharing. First, a secret data item D is divided into n shares $D_1,...,D_n$. Then, a threshold k is chosen so that: (a) to recover D, k or more of arbitrary D_i's are required; (b) using any $k-1$ or fewer D_i's leaves D completely undetermined.

Ben-Or, Goldwasser and Wigderson [15] define a protocol for multi-party computing of a function f using secret input from all the parties. The protocol involves n "regular" parties, which calculate only partial function outputs. In [15], one of the players is selected as the *dealer* (denoted by us as *DLR*), and is provided the partial function outputs to find out the full results of function computation. Let f be a *linear function* of degree n known to each of the n parties, and t be an arbitrary threshold value. Let P_i denote Party i, and x_i denote the *secret* input of P_i for f. Dealer *DLR* will receive from the n parties the partial outputs of f calculated by the n parties using their respective secret inputs $x_1, x_2, \ldots, x_n$. Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be distinct non-zero elements in the domain of f. Player P_i is assigned the point α_i.

Each party P_i generates a polynomial h_i of degree t (where t is the above threshold value) such that $h_i(0) = x_i$. Each P_i sends to each P_j (from the subset of the other $n-1$ parties) one share $s_{i,j} = h_i(\alpha_j)$ of P_j's input. Then, each P_i computes a portion of function f using shares $s_{i,j}$ of the input that it has (its own) or received from $n-1$ other parties.

A predicate encryption scheme allows evaluating predicates with encrypted data. For example, Alice can compute the predicate "(email sender = 'Bob') and (date in [2006, 2007])" using encrypted data [13].

Fig. 3 shows a sample predicate encryption scheme that has this property [13]. Alice uses a *Setup* algorithm to generate a public key *PK* and a secret key *MSK*. Next, Alice uses *PK* to encrypt (with algorithm *Encrypt*) her PII and gets ciphertext *CT*. Then, she can store *CT* (the encrypted *PII*) on an untrusted host (e.g., in a cloud). She may also publish *PK*, so that it can be used to encrypt data that she can access.

Alice has the function p representing a predicate that she wishes to evaluate for her encrypted PII. She uses the *KeyGen* algorithm, *PK*, *MSK* and p to output the token TK^p (encoding p). Then, she gives TK^p to the host that evaluates the token (with p included in the token) for *CT* (the encrypted PII), and returns the result *p(PII)* to Alice.

Note that *KeyGen* uses the secret key *MSK* as input. Hence, Alice can use *KeyGen* to generate TK^p for p. Alice can give TK^p to an untrusted host while protecting PII. (Observe that if Alice gave *KeyGen* and *MSK* to the host, the scheme would not be secure—it would not protect PII.)

1.	Setup	*PK*, *MSK*
2.	Encrypt(*PK*, PII)	*CT*
3.	KeyGen(*PK*, *MSK*, *p*)	TK^p
4.	Query(*PK*, *CT*, TK^p)	*p*(PII)

Figure 3. The Public-key Predicate Encryption Scheme

For negotiating use of a cloud service, we combine computing predicate over encrypted data with secure multi-party computing. The secret key *MSK* is split between n parties [1] using the above-mentioned Shamir's technique. Then, the algorithm *KeyGen* is provided to n parties, and computed by them collaboratively using their shares of the secret key, function p representing a predicate, *PK*, and *TK*. This is done as specified in the above-mentioned protocol of Ben-Or, Goldwasser and Wigderson for multi-party computing.

In our algorithm, an owner O encrypts PII using algorithm *Encrypt* and O's public key *PK*. *Encrypt* outputs *CT*—the encrypted PII. SP transforms his request for PII to a predicate represented by function p. Then, *SP* sends shares of p to the n parties who hold the shares of *MSK*. The n parties execute together *KeyGen* using *PK*, *MSK*, and p, and return TK^p to SP. Next, SP calls the algorithm *Query* that takes as input *PK*, *CT*, TK^p and produces *p*(*PII*) which is the evaluation of the predicate. The owner O is allowed to use the service only when the predicate evaluates to "true".

[1] The decryption key shares are stored in a distributed hash table system, such as the one proposed by Vuze [18], where each share is stored in a different location.

B. Use of Active Bundle Scheme for IDM

Let us take a look now at the active bundle scheme, and its use for IDM.

B.1. Overview of the Active Bundles Scheme

An *active bundle* includes sensitive data, metadata, and a virtual machine [16]. Sensitive *data* contains content to be protected from privacy violations, data leaks, unauthorized dissemination, etc.—e.g., it contains PII.

Metadata describes the active bundle and its privacy policies. The metadata includes (but is not limited to) the following components (details available in [16]): (a) *integrity check* metadata; (b) *access control* metadata; (c) *dissemination control* metadata; (d) other application-dependant and context-dependant metadata.

Virtual machine (VM) manages and controls the program code enclosed in a bundle. Its main functions include: (a) enforcing bundle access control policies through apoptosis, evaporation, or decoy actions; (b) enforcing bundle dissemination policies; and (c) validating bundle integrity.

B.2. Using Active Bundles for IDM

The components of an active bundle for IDM are:

1) *Identity data:* Data used for authentication, getting service, using service (e.g., SSN, DOB). These data are encrypted and packed inside the active bundle.
2) *Disclosure policy:* A set of rules for choosing which identity data to disclose. E.g., if identity data *I* are used for service *S*, then *I* should be used each time *S* is accessed (minimizing disclosure of PII).
3) *Disclosure history:* Used for logging and auditing purposes. It is also used for selecting identity data to be disclosed based on previous disclosures.
4) *Virtual Machine:* It contains the code/algorithm for protecting PII, on untrusted hosts.

An active bundle is sent from a source host to a destination host. When arriving at a "foreign" host, an active bundle ascertains the host's trust level through a TTP [16]. Using its disclosure policy, it decides whether the host may be eligible to access all or part of bundle's data, thus becoming a "guardian" for the data, and which portion of sensitive data can be revealed to it. The remaining data (not to be revealed) might be *evaporated* as specified in the access control policies, protecting the data. We consider a number of different metrics for adaptive control of the degree of evaporation, including trust-based metrics.

An active bundle may realize that its security is about to be compromised. E.g., it may discover that its self-integrity check fails, or the trust level of its host is too low. In response, the bundle may choose to apoptosize, that is, perform atomically a clean self-destruction (one that is complete and leaves no traces usable for an attacker) [16].

We developed a prototype for the active bundle scheme. Fig. 4 shows an example of the activation process for a generic-purpose bundle. Fig. 4a displays the log of Security Server Agent (SSA), a major component of the Active Bundle Prototype. The encircled text shows SSA sending a message to a bundle warning it that host's trust level is 4 (while the trust threshold for the bundle is 5). Fig. 4b displays the log of the bundle indicating apoptosizing.

III. Advantages of the Proposed Approach

The approach presented in this paper is one of the alternatives to using TTPs. It reduces the risks associated with the use of TTPs. The main advantages of the proposed approach are:

1) *No need for TTPs*. Since data exchange between a bundle and its host is local to the host, it protects PII from man-in-the-middle, side channel and collaborative attacks.
2) *Authentication without disclosing unencrypted data*. This prevents unnecessary data disclosures.
3) *Protection of identity data from untrusted hosts*. If data reach an unintended destination or are tampered with, they self-destroy by apoptosis or evaporation to prevent falling into wrong hands.

```
SecurityServerBehavior--securityagent received msgcontent: ((action (age
nt-identifier :name ActiveBundle@ABFramework) (ABIdentityItem :ABName Ac
tiveBundle2 :ABPublicKey "" :ABRole myrole :ABRequiredTrust 5 :ABHostTru
st 0)))
SecurityServerBehavior--conversationID:RegisterABidentity
SecurityServerBehavior-- Message RegisterABidentity received from ( agen
t-identifier :name ActiveBundle2@ABFramework  :addresses (sequence http:
//css01.cs.wmich.edu:7778/acc http://css01.cs.wmich.edu:34403/acc http:/
/css01.cs.wmich.edu:35004/acc http://css01.cs.wmich.edu:55033/acc ))
ABidentity--key[B@24c68a98,Algorithm:DES,format:RAW
KeyManager--Write identity of ActiveBundle2 to file ABkeys.obj
KeyManager--Identities file is saved
SecurityServerAgent-- RegisterABidentity performed on identity of ( agen
t-identifier :name ActiveBundle2@ABFramework  :addresses (sequence http:
//css01.cs.wmich.edu:7778/acc http://css01.cs.wmich.edu:34403/acc http:/
/css01.cs.wmich.edu:35004/acc http://css01.cs.wmich.edu:55033/acc ))
SecurityServerBehavior--RegisterABKey
SecurityServerBehavior--securityagent received msgcontent: ActiveBundle2
SecurityServerBehavior--conversationID:RequestDecryptionInformation
SecurityServerBehavior--ab ActiveBundle2
SecurityServerBehavior--send message  My trust?
TrustServerBehavior--TrustServerBehavior received msgcontent: My trust?
------------------------
TrustServerBehavior--reply with trust value 4
Time: 2010/07/24 18:01:11 Active Bundle: ActiveBundle2 Source:141.218.14
3.147 Current: 141.218.143.147
SecurityServerBehavior-- Received from trust server -- 4
trust4
AuditAgentBehavior-- Message  received  ABAuditItem@b0c0f66
SecurityServerBehavior--send reply RequestDecryptionInformation-- ((acti
on (agent-identifier :name securityServer@ABFramework) (ABIdentityItem :
ABName ActiveBundle2 :ABPublicKey "" :ABRole myrole :ABRequiredTrust 5 :
ABHostTrust 4)))
```

a) Log of Security Server Agent during Activation of an Active Bundle

```
ActiveBundleActivateBehavior-----ActiveBundleActivateBehavior --Action
ActiveBundleActivateBehavior--Send Audit information about ActiveBundle2 to
 Audit Agent
ActiveBundleActivateBehavior--Send a message to the security server Request
DecriptionInformation
ActiveBundleActivateBehavior--reply: ABIdentityItem@57ac3379
ActiveBundleActivateBehavior-- Apoptosis due to trust level
ActiveBundleActivateBehavior-----Apoptosis for active bundle ActiveBundle2@
ABFramework on host:141.218.143.147
ActiveBundleActivateBehavior-----Deletion Done ...
ActiveBundleActivateBehavior--Activation done...
ActiveBundleAgent-- Agent shutdown
ActiveBundleAgent--ActiveBundleAgent ActiveBundle2@ABFramework terminating.
```

b) Active Bundle Log Showing its Apoptosis during its Activation

Figure 4. Logs for Security Server Agent and Active Bundle.

IV. Resilience of the Proposed Approach to Attacks

A system based on the proposed approach is independent of the usage of TTP. This reduces the risks of correlation attacks within the cloud.

Correlation attacks on IDM happen when an entity acquires a set of data (multiple PIIs in case of IDM) and is

able to correlate it to the physical identify of an entity such as a person. Approaches that use a TTP increase the risk of correlation attacks on an entity's PII. Approaches that do not use a TTP reduce the risk of such attacks.

Ristenpart *et al.* [17] demonstrated that Amazon cloud is prone to side-channel attacks and it would be possible to steal data, once the malicious virtual machine is placed on the same server as its target. It is possible to carefully monitor how access to resources fluctuates and thereby potentially glean sensitive information about a victim. Though, they point out, that there are a number of factors that would make such an attack significantly more difficult in practice.

Approaches that use a TTP increase the risk of side-channel attacks on an entity's PII. Approaches that do not use a TTP such as the one that we use reduce the risk of such attacks.

The proposed solution is prone to other attacks. E.g., the active bundle may also be not executed at all at the host of the requested service. In this case its data is not disclosed but the user is denied access to the service that he requests.

V. CONCLUSION

With the immense growth in the popularity of cloud computing, privacy and security have become important concerns for both the public and private sectors. It is very likely that users end up having multiple identities in multiple service providers' (SPs) security repositories, as well as multiple credentials and multiple access permissions for different services provided by different SPs.

There is a strong need for an efficient and effective privacy-preserving system that is independent of TTPs, able to unambiguously identify users that can be trusted both within enterprises and across the Web, and protects users' PII. IDM is one of the core components in cloud privacy and security, and can alleviate some problems associated with cloud computing.

We propose an approach for building IDM systems without using TTPs, using the active bundles scheme, computing predicate over encrypted data and multiparty computing. The solution allows the use the IDM application on untrusted hosts.

Future work will include development of a prototype for the proposed IDM system for cloud computing, and testing it for diverse real-world scenarios. The goal is to prove effectiveness and efficiency of the proposed approach.

VI. ACKNOWLEDGEMENT

This research was supported by a contract from AFRL and NGC Corp.

REFERENCES

[1] "Cloud Computing," NIST, accessed in Aug. 2010. Online at: http://csrc.nist.gov/groups/SNS/cloud-computing/

[2] R. Gellman, "Privacy in the Clouds: Risks to Privacy and Confidentiality from Cloud Computing," *World Privacy Forum,* Feb. 2009. Online at: http://www.worldprivacyforum.org/pdf/WPF_Cloud_Privacy_Report.pdf

[3] P. Angin, B. Bhargava, R. Ranchal, N. Singh, L. Ben Othmane, L. Lilien, and M. Linderman, "A User-Centric Approach for Privacy and Identity Management in Cloud Computing," *Proc. 29th IEEE Intl. Symp. on Reliable Distributed Systems* (*SRDS*), New Delhi , India, Nov. 2010.

[4] A. Gopalakrishnan, "Cloud Computing Identity Management," SETLabs Briefings, vol. 7, 2009. Online at: http://www.infosys.com/research/

[5] OpenID Foundation Website, accessed in Aug. 2010. Online at: http://openid.net/

[6] K. Cameron, "Identity Web blog," accessed in Aug. 2010. Online at: http://www.identityblog.com/?=p685

[7] *S. Fischer-Hubner*, and H. Hebdom, "PRIME - Privacy and Identity Management for Europe," accessed in Aug. 2010. Online at: https://www.prime-project.eu/prime_products/reports/fmwk/pub_del_D14.1.c_ec_wp14.1_v1_final.pdf

[8] L. Lilien, and B. Bhargava, "A Scheme for Privacy-preserving Data Dissemination," *IEEE Trans. on Systems, Man and Cybernetics*, Part A: Systems and Humans, Vol. 36(3), 2006.

[9] J. Falkner, M. Piatek, J. John, A. Krishnamurthy, and T. Anderson, "Profiling a Million User DHT," *Proc. 7th ACM SIGCOMM Conference on Internet Measurement.* San Diego, CA. Oct. 2007, pp. 129-134.

[10] K. Cameron, "Proposal for a Common Identity Framework: A user-Centric Identity Metasystem," June 2009. Online at: http://www.identityblog.com/?p=1048

[11] W.A. Alrodhan, and C.J. Mitchell, "Improving the Security of CardSpace," *EURASIP Journal on Information Security*, vol. 2009, 2009, doi:10.1155/2009/167216

[12] Whoops dept., "AT&T Security Hole Revealed Email Addresses Of iPad Owners," June 2010. Online at: http://www.techdirt.com/articles/20100609/1604379757.shtml

[13] E. Shi, "Evaluating Predicates over Encrypted Data," Ph.D. Thesis. Carnegie Mellon University, Pittsburgh, PA. Oct. 2008.

[14] A. Shamir, "How to Share a Secret," *Communications of the ACM*, vol. 22(11), Nov. 1979, pp. 612–613.

[15] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," *Proc. Twentieth Annual ACM Symposium on Theory of Computing.* Chicago, IL. May 1988, pp.1-10.

[16] L. Ben Othmane, and L. Lilien, "Protecting Privacy in Sensitive Data Dissemination with Active Bundles," *Proc. 7th Annual Conference on Privacy, Security & Trust* (*PST 2009*), Saint John, New Brunswick, Canada, Aug 2009.

[17] T. Ristenpart, E. Tromer, H. Shacham, S. Savage, "Hey, You, Get Off My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," *Proc. 6th ACM conference on Computer and Communications Security*, Chicago, IL, Nov. 2009, pp. 199-212.