ACCOUNTABLE REPUTATION RANKING SCHEMES FOR SERVICE PROVIDERS IN CLOUD COMPUTING

Wassim Itani, Cesar Ghali, Ayman Kayssi and Ali Chehab
Department of Electrical and Computer Engineering, American University of Beirut, Beirut, Lebanon

Keywords: Reputation, Secure event logs, Cryptographic coprocessors, Cloud computing, Secure performance monitoring.

Abstract: We present RaaS (Reputation as a Service), a set of accountable reputation ranking schemes for service providers in cloud computing architectures. RaaS provides a secure reputation reporting system producing results and recommendations that can be published as a service and verified by trusted third parties or by the cloud service providers themselves. The reputation service is based on an assortment of ranking criteria ranging from multilevel performance and quality of service measures to security and pricing assessments. This makes RaaS a valuable IT component in supporting verifiable and accountable compliance with service-level agreements and regulatory policies, encouraging competition among cloud providers for better security and quality of service, and providing new and existing cloud customers with valuable advice for selecting the appropriate cloud service provider(s) that suit their performance, budgeting, and security requirements. The RaaS reputation system does not rely on subjective feedback from cloud customers but rather carry out the reputation calculation based on observable actions extracted from the computing cloud itself. A proof of concept implementation shows that the incorporated RaaS protocols impose minimal overhead on the overall system performance.

1 INTRODUCTION

Cloud computing has achieved unprecedented success and adoption in the last few years. This evolutionary computing model relies on the great advancements in virtualization technologies, commodity hardware, processor design, and most importantly Internet communication networks to provide compelling services to enterprises and individuals.

Currently, some cloud service providers guarantee the quality of their services by defining a set of Service Level Agreements (SLAs) with their customers. These SLAs typically lack any technical means of enforcement which leaves the customer's data and software processes under the total control of the cloud service provider. Any failure to meet the SLA terms and obligations will have disastrous effects on the cloud customer and provider, such as losing reputation and client trust and legal or financial penalties that may lead to putting an end to the entire business. This fact will put pressure and responsibility on the customers when selecting a particular cloud service provider for running their business processes and storing data. The severity of this decision is further aggravated when we estimate the serious losses incurred when dealing with "misbehaving" cloud providers or the technical difficulties, financial losses, and service downtimes accompanying the process of switching between service providers. Terabytes of data migration tasks over expensive communication links, software reconfiguration and adaptation, and data leakage and privacy implications are some factors that render the migration process highly expensive.

To alleviate customers concerns, and to aid them in selecting the appropriate cloud service provider at the outset, we believe that a secure and accountable cloud reputation service should be developed to rank service providers based on performance, security, and pricing criteria. The advantages of such a reputation service would be reflected on both customers and providers. The cloud customer will be able to take better selection decisions when choosing a cloud infrastructure guided by a set of measurable and quantified reputation scores. On the other hand, the reputation service will encourage the cloud

Itani W., Ghali C., Kayssi A. and Chehab A..
ACCOUNTABLE REPUTATION RANKING SCHEMES FOR SERVICE PROVIDERS IN CLOUD COMPUTING.
DOI: 10.5220/0003384100490055
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 49-55
ISBN: 978-989-8425-52-2

provider to enhance its provided services and offerings to attract a larger customer base.

In this paper, we present RaaS, a set of accountable reputation ranking schemes for service providers in cloud computing architectures. RaaS provides a secure reputation reporting system producing results and recommendations that can be published as a service and verified by trusted third parties or by the cloud service providers themselves. The reputation service is based on an assortment of ranking criteria ranging from multilevel performance and quality of service measures to security and pricing assessments. This makes RaaS a valuable IT component in supporting verifiable and accountable compliance with service-level agreements and regulatory policies, encouraging competition among cloud providers for better security and quality of service, and providing new and existing cloud customers with valuable advice for selecting the appropriate cloud service provider(s) that suit their performance, budgeting, and security requirements. The RaaS reputation system does not rely on subjective feedback from cloud customers but rather carry out the reputation calculation based on observable actions extracted from the computing cloud itself. A proof of concept implementation shows that the incorporated RaaS protocols impose minimal overhead on the overall system performance.

The rest of this paper is organized as follows: in Section 2 we discuss the main system players assumed in this work. Section 3 provides a brief overview of a proposed hardware-based security infrastructure in cloud computing. Section 4 presents the design of the RaaS reputation system. In Section 5, a proof of concept implementation of the RaaS protocols is evaluated and analyzed. Section 6 provides a literature survey of the main models related to the proposed work. Conclusions are presented in Section 7.

2 RaaS MAIN PLAYERS

RaaS operates in a traditional cloud computing environment consisting of the two main communicating entities, namely a cloud service provider and a cloud customer. The provider manages and operates a cloud infrastructure of on-demand storage and processing services. The customer consumes the cloud services provided by the service provider.

The cloud applications managed by the provider abide with the 3-tier enterprise architecture. In this model, the enterprise application is divided into three main logical layers. The first layer is the presentation layer which is responsible for providing the application's user interface. The second layer, the business layer or the middle tier, is responsible for executing the application's business logic. Finally, the third layer, the data persistence layer, is responsible for storing and maintaining the data required by the cloud application (mainly in relational database management systems).

3 HARDWARE-BASED SECURITY IN THE CLOUD: A PROPOSED TECHNICAL OVERVIEW

For a reputation system to be trustworthy, it has to rely on a trusted cloud computing infrastructure that supports the accountability and credibility of the overall system operation. RaaS relies on secure cryptographic coprocessors to enforce the accountability of the reputation calculation protocols. This section is inspired from our previous work on privacy-aware data storage and processing in cloud computing (Itani, Kayssi, Chehab, 2009).

3.1 Secure Coprocessor Overview

A cryptographic coprocessor is a tamper-proof piece of hardware that interfaces, mainly via a PCI-based interface, to a main computer or server. The chief security property supported by a crypto coprocessor is its ability to provide a secure and isolated execution environment in the computing cloud. A crypto coprocessor is a full-fledged computing system with its own processor, RAM, ROM, battery, network interface card, and persistent storage. However due to economic reasons, the coprocessor is usually less capable in terms of processing and memory resources than the main server it interfaces to, and thus cannot replace it.

3.2 Coprocessor Authoritative Configuration and Distribution

The main authoritative entity responsible for configuring the crypto coprocessors and distributing them to cloud service providers is a trusted third-party (TTP) trusted by the cloud provider and customer. The resources of the crypto coprocessors installed in the computing cloud are shared among the cloud customers registered in the provider’s storage and processing services. This resource sharing mechanism supports the economic feasibility of the solution and, most importantly, complies with the general cloud computing vision and paradigm. The TTP is responsible for securing the provider rating process and analyzing the rating records to generate the provider's absolute reputation score. Technically, the TTP loads a set of private/public key pairs into the persistent storage of the crypto coprocessor. Every public/private key pair (PU_{CID}/PR_{CID}) is to be securely allocated and distributed to a single customer when the latter registers with the cloud service provider.

In addition to loading the customer's PU_{CID}/PR_{CID} key pair, the TTP also loads its own secret key K_{TTP} into the persistent storage of the crypto coprocessor. This key is needed by the TTP to remotely authenticate to the crypto coprocessor and to securely execute commands against it. Moreover, K_{TTP} is used to secure the integrity and confidentiality of log records produced by the RaaS reputation protocols on the cloud provider's side.

The notion of a TTP has proved to be feasible in certification authority services in public-key infrastructures (PKIs).

3.3 Coprocessor Process Model and Software Division

The crypto coprocessor process structure abides by the ABYSS (Weingart, 1987) model. RaaS supports the economic and performance feasibility of the reputation system by adopting the software division concept. This concept urges the cloud customer to logically divide its software application components into two categories: protected and unprotected. The protected classification indicates that the logical component should be executed in a protected process in the address space of the cryptographic coprocessor. On the other hand, the unprotected classification indicates that the logical component can be executed in a traditional process on the main server's processor.

The software division mechanism is also applied to the RaaS performance evaluation protocols to ensure the accountability of the feedback logging mechanism on the cloud provider's side. In the same sense, the protocols execution steps are divided into secure and non-secure based on their role in the performance evaluation process. It is worth mentioning that it is possible to run the entire software application in the secure coprocessor; however, this would affect the performance and economic efficiency of the application.

4 REPUTATION SYSTEM ARCHITECTURE

RaaS provides a secure and accountable reputation service that does not rely on subjective feedback from cloud customers. The main source of input feeding the reputation calculation mechanism resides in the computing cloud itself at the provider side. The reputation service utilizes a trusted crypto coprocessor to provide a secure execution environment in the computing cloud, and thus produces authentic event logs that constitute the basis for the reputation score calculation.

The provider's reputation score is computed by the TTP and consists of three main components: a security reputation score, a pricing reputation score, and a performance reputation score which is subdivided into two sub scores: the data retrieval and processing reputation scores.

The reputation service consists of three main phases: the secure event monitoring and auditing phase, the reputation score calculation phase, and the service publication phase.

4.1 Secure Event Monitoring and Auditing

Depending on the type and sensitivity of operations requested by the cloud customer, three performance evaluation and logging protocols are developed to securely generate the event logs about customer transactions. The RaaS performance evaluation protocols are presented below:

4.1.1 The Bulk Data Fetch (BDF) Protocol

This protocol is executed whenever the cloud client sends a query for fetching bulk database/file data from the cloud storage facility. The main goal behind running this protocol is to securely and accurately measure the time needed by the cloud service provider to retrieve the requested data from the cloud storage. Figure 1 presents the interaction between the trusted coprocessor and the main server processor to execute the BDF protocol steps. In step 1, the cloud customer sends a storage query along with its authentication credentials to the computing cloud. In step 2, the crypto coprocessor authenticates the customer and initiates a performance timer at time t_1. In step 3, the crypto coprocessor relays the query to the main server processor which fetches the required data from the storage facility in steps 4 and 5. In step 6, the main server processor calculates and sends to the crypto coprocessor a hash of the

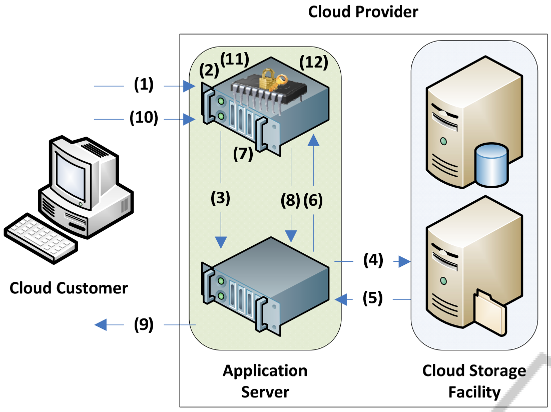

Figure 1: The BDF protocol execution steps.

retrieved data. The hash message has a double purpose; firstly it represents a signaling message from the main server processor to the crypto coprocessor indicating that the data fetch process is accomplished. Secondly, the hash value constitutes a commitment that binds the cloud service provider with the results it fetched from the cloud storage. This commitment scheme prevents the cloud provider from rushing a fake and premature signal to the secure coprocessor before the actual data fetching mechanism is really executed. In step 7, the crypto coprocessor terminates the performance timer at time t_2 and verifies the received hash value, and then it authorizes the main server processor to deliver the data to the cloud customer in steps 8 and 9. In step 10, the client sends the hash value of the query data received to the crypto coprocessor. In step 11, the crypto coprocessor compares the commitment hash value received from the main server processor in step 6 with the hash value received from the client in step 10. Equal hash values indicate that the cloud provider has fully accomplished the query request before sending the finish signal to the crypto coprocessor in step 6. Finally in step 12, the crypto coprocessor generates a secure log entry containing, most importantly, the value of the time interval ($t_2 – t_1$).

4.1.2 The Data Fetch for Execution (DFE) Protocol

This protocol is a variant of the BDF protocol with the exception that the fetched data is not sent to the cloud customer; instead it is consumed by internal software processes. If the software processes are secure, i.e. running in the address space of the crypto coprocessor, then steps 8, 10, 11, and 12 will not be needed since in this case the processes receiving the data are already running on a trusted platform. Step 9 will be modified to deliver the requested data to a software process instead of the cloud customer. This will be illustrated in the next section.

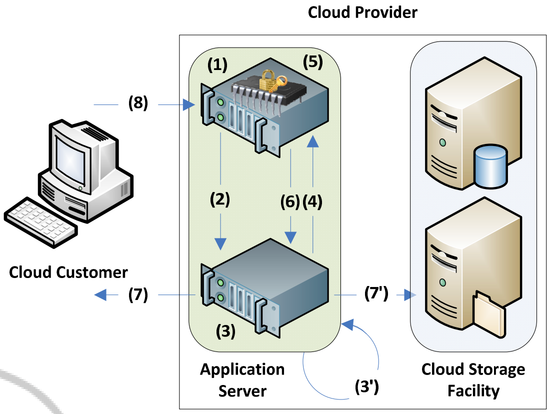

Figure 2: The DEM protocol execution steps.

4.1.3 The Data Execution Monitoring (DEM) Protocol

This protocol is executed after the DFE protocol. Its chief goal is to measure the time needed by the internal software processes on the cloud provider side to process the data retrieved by the DFE protocol. A very important property of the DEM protocol is that it operates on non sensitive customer data. Processing operations on sensitive data do not necessitate the presence of a dedicated performance evaluation protocol. This fact is illustrated as follows: Due to the high sensitivity of the processed data, the software processes handling it should be of the secure type; that is running in the address space of the secure crypto coprocessor. Since the processing platform is trusted and controlled by the TTP, no performance evaluation is carried out or required and thus this protocol has no direct effect on the reputation score calculation. Figure 2 illustrates the DEM protocol execution steps. In step 1, the crypto coprocessor identifies the start of the data processing by initiating the performance timer at time t_1, and commands the main server processor to start data processing in step 2. In step 3, the main server processor carries out the processing task and produces the final execution results (execution pipelining may be employed here). In step 4, the server main processor sends a hash of the final result to the secure coprocessor. The purpose of this hash value is analogous to that described in the BDF protocol step 6. In step 5, the secure coprocessor terminates the timer at time t_2 and validates the received hash value, and then it authorizes the main server coprocessor to send the results to the cloud customer or to store them in the cloud storage facility in steps 7 and 7′, respectively. In step 8, the cloud customer sends the hash of the processing results, received in step 7, to the crypto coprocessor. The secure processor validates the commitment hash values and then generates a secure log entry in steps 9 and 10.

4.2 Reputation Calculation

This section describes the performance, security, and pricing reputation scores calculation. The reputation scores for providers are calculated periodically by the TTP which can carry out the calculation on a monthly, quarterly, semi-annual, or annual basis.

4.2.1 Performance Reputation Scores

The securely generated logs are analyzed in order to calculate the provider performance reputation scores. The calculation is done as follows: for each log entry, the TTP utilizes the τ (the data size in bytes retrieved in the transaction) and β (the time required to retrieve the data in the transaction in milliseconds) fields to calculate the transaction data retrieval rates and compares them to the rates promised in the SLA based on the following equations:

$$\Phi_{\mathrm{r},i}\,[\%] = \frac{\frac{\beta_i}{\Psi_\mathrm{r}} - \tau_i}{\frac{\beta_i}{\Psi_\mathrm{r}}} \times 100 \qquad (1)$$

$\Phi_{\mathrm{r},i}$ represents the percent improvement over the SLA retrieval rate Ψ_r for the i^{th} log record. By applying averaging and normalization operations on the calculated Φ_r log entry improvement rate, the overall retrieval reputation score R_r can be computed. R_r is a function of the average retrieval rates $\bar{\mathrm{X}}_\mathrm{r}$ and their standard deviations σ_r.

$$\bar{\mathrm{X}}_\mathrm{r}\,[\%] = \frac{1}{N_r} \times \sum_{i=1}^{i=N_r} \Phi_{\mathrm{r},i} \qquad (2)$$

$$\sigma_\mathrm{r}\,[\%] = \sqrt{\frac{1}{N_r} \times \sum_{i=1}^{i=N_r} \left(\Phi_{\mathrm{r},i} - \bar{\mathrm{X}}_\mathrm{r}\right)^2} \qquad (3)$$

$$R_\mathrm{r} = \left[\lambda + \left(\lambda \times \frac{\bar{\mathrm{X}}_\mathrm{r}\,[\%]}{100}\right)\right] \pm \left(\lambda \times \frac{\sigma_\mathrm{r}\,[\%]}{100}\right) \qquad (4)$$

Where N_r is the number of retrieval log entries per provider, and λ is a normalization constant that represents the middle value of the reputation score range. The range of the retrieval reputation score component lies in $]0, 10[$, hence $\lambda = 5$. Note that the data processing reputation score R_p is calculated analogously.

4.2.2 Security Reputation Scores

The security reputation calculation in RaaS is carried out in two main phases: a static analysis phase and a dynamic penetration testing phase. In the static analysis phase, the TTP statically analyzes the provider's SLAs for security-related terms and specifications, classifies them into a set of security categories, and assigns a reputation weight to each category. The category reputation weight is provided based on the quality of security this category represents.

Three security categories are currently supported in RaaS: (1) The degree of provider's compliance with regulatory policies and recommendations, (2) the set of cryptographic protocols supported by the cloud service provider, and (3) the strength of the symmetric and asymmetric keys used in these cryptographic protocols. The list of security categories is implementation dependent and can be extended with additional classification groups as devised by the TTP.

In the dynamic penetration testing phase the TTP performs a security assessment of the provider's site using advanced and up-to-date vulnerability scanning techniques such as the Nmap and Nessus network scanners. The dynamic penetration testing phase aims at scanning the provider's network resources and applications for known vulnerabilities to verify the immunity of the provider's system against possible security exploitations.

4.2.3 Pricing Reputation Scores

The pricing reputation score is calculated by ranking a set of service providers according to the cost of the cloud services they provide. The ranking is achieved by employing a simple order statistics algorithm. In brief terms, the algorithm assigns higher pricing scores to providers offering lower service prices.

4.3 Reputation Publication

After the TTP accomplishes the log analysis and reputation score calculation, it publishes the results online as a cloud service. The TTP also provides a set of procedures for resolving disputes and enabling the cloud service provider to check the validity and coherency of the provider published reputation scores.

5 RaaS PROTOTYPE IMPLEMENTATION

A prototype proof of concept of the RaaS reputation service algorithms and protocols is implemented on the VMware vSphere 4 cloud computing platform. We created five client virtual machines (VMs) on the vSphere virtualization server to support the execution of the customer application business logic. The guest operating systems running on these machines are: 2 Windows XP SP3 VMs, 1 Windows 7 VM, and 1 Ubuntu 9.04 VM. The vSphere physical server specifications are as follows: Intel(R) Core(TM) i7 CPU Q 720 running at 1.6 GHz equipped with 4GB RAM. We implemented 2 sample customer enterprise applications, using the C# programming language, to run in the vSphere cloud: A Customer Relationship Management (CRM) application and a Human Resource (HR) management application. The applications execute SQL queries on an SQL Server 2005 RDBMS. To implement the functionality of a secure crypto coprocessor, we assume that one of the core CPUs on the virtualization server is the secure coprocessor while the other core CPUs are those of the main untrusted server. We believe this assumption provides a viable proof of concept sufficient for testing the system configuration, functionality, and reputation mechanisms.

For evaluating the performance reputation score, a set of 2000 data retrieval and processing transaction events is generated by the implemented performance monitoring protocols. The event records generated are evenly distributed over 4 virtual time periods. The data size, retrieved or processed, ranges from 10 KB to 10 MB. The transaction logs are analyzed and processed based on the equations presented in Section 4.2 to produce the RaaS performance reputation scores.

Employing the RaaS performance monitoring protocols and secure log generation mechanisms added minimal overhead to the overall application performance. This fact is illustrated in Figures 3 which presents the average time in seconds consumed by the data retrieval and processing operations with and without the application of the RaaS performance monitoring protocols. The overhead is roughly 15% for the different RaaS performance evaluation protocols. We believe that this cost is considered reasonable in return of the reputation service provided.

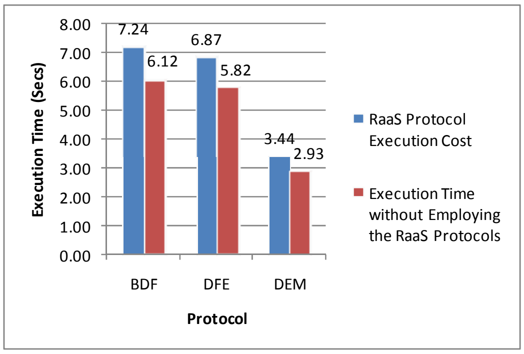

Figure 3: Execution time with and without the application of the RaaS performance protocols.

6 RELATED WORK

A common property shared by existing service-oriented reputation systems is that they base the reputation calculation on the consumers' feedback. Since this form of feedback information maybe, in many cases, subjective, biased, or even malicious, the results and recommendations provided by this category of reputation systems is characterized by incompleteness and inaccuracy and thus cannot be fully trusted. Mármol and Pérez (2009) present some of the key challenges and threats facing the process of reputation calculation in distributed systems. According to (Mármol and Pérez, 2009), differentiating among honest and dishonest clients and handling malicious peers and information collectives are on top of the list of challenges and risks to be tackled when designing distributed reputation systems. Some of the proposed service-oriented reputation systems are presented in (Malik and Bouguettaya, 2009; Chang, Dillon and Hussain, 2006; Hwang, Kulkareni and Hu, 2009). A comprehensive survey of Internet trust and reputation systems is presented in (Lim, Keung and Griffiths, 2010).

Haeberlen (2009) discusses the key requirements for establishing an accountable computing cloud and suggests the presence of an "Audit" primitive function that enables the customer to check the compliance of the provider with the service agreements. The requirements provided in (Haeberlen, 2009) are not accompanied with a technical solution. This paper is viewed as a "call for action" for further research in this field as stated by the author.

Li, et al. (2010) present a set of benchmarking tools for estimating the performance and costs of deploying a customer cloud application on different cloud providers. This work suffers from a set of limitations that RaaS overcomes by design: (1) they do not consider the validity of the benchmarking results when possibly dealing with malicious cloud providers, (2) the performance measurements produced represent a snapshot in time and hence they are affected by variations in customer's workloads or by any modification in the software, hardware, or network infrastructure, (3) they represent a client-side estimate of the provider's performance and (4) they do not consider any security evaluation metric which we believe is a major requirement that should be considered when selecting a cloud provider.

A considerable amount of research work has dealt with the design and implementation of secure cryptographic coprocessors. The secure crypto coprocessor concept was firstly introduced by Best (1980). The advancements in physical security mechanisms and packaging technology (Weingart, 1987) and the assortment of secure applications that can be implemented on top of physically secure coprocessors (Tygar and Yee, 1994) was a major driving force to a prosperous commercial market. IBM was the leader on this front by providing a set of successful implementations meeting the strictest FIPS 140 security standards. This is represented in the IBM 4758 PCI cryptographic coprocessor (Dyer, et al., 2001) and the IBM 4764 PCIX cryptographic coprocessor (PCIXCC) (Arnold and Doorn, 2004). The IBM coprocessor product family was the first to meet the FIPS level 4 security standard based on its tamper-resistance and tamper-responding mechanisms. Moreover, Gutmann (2000) presented a general-purpose open-source crypto coprocessor that provides competitive performance and higher functionality compared to commercial products at a cost of one to two orders of magnitude lower.

7 CONCLUSIONS AND FUTURE EXTENSIONS

In this paper we presented RaaS, a set of accountable reputation ranking schemes for service providers in cloud computing architectures. RaaS builds on a set of integrity-assurance mechanisms and protocols to provide a secure execution environment for supporting the reputation calculation. Dedicated light-weight performance evaluation protocols are established to secure the event log generation and storage mechanisms. A prototype implementation of the various RaaS algorithms and protocols is tested on the VMware vSphere 4 cloud computing operating system. The incorporation of the RaaS protocols added negligible overhead to the overall system performance.

Future extensions will include: augmenting a more comprehensive description of the reputation protocols, devising a cumulative reputation score calculation mechanism, and extending the system simulation with a set of stochastic load and stress factors.

REFERENCES

Itani, W., Kayssi, A. and Chehab, A., 2009. Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures. In *DASC'09*.

Tygar, J. and Yee, B., 1994. Dyad: A system for using physically secure coprocessors. In *IP Workshop*.

Weingart, S., 1987. Physical security for the mABYSS system. In *IEEE Computer Society Conf. on Security and Privacy*.

Schneier, B. and Kelsey, J., 1999. Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security*, 1(3), pp.159-196.

Mármol, F. and Pérez, G., 2009. Security threats scenarios in trust and reputation models for distributed systems. *Computers and Security*, 28(7), pp.545-556.

Malik, Z. and Bouguettaya, A., 2009. RATEWeb: reputation assessment for trust establishment among web services. *VLDB Journal*, 18(4), pp.885–911.

Chang, E., Dillon, T. and Hussain, F., 2006. Trust and reputation for service-oriented environments. Wiley.

Hwang, K., Kulkareni, S. and Hu, Y., 2009, Cloud Security with Virtualized Defense and Reputation-Based Trust Management. In *DASC'09*.

Lim, S., Keung, C. and Griffiths, N., 2010. Trust and Reputation. In *Springer Agent-Based Service-Oriented Computing*.

Haeberlen, A., 2009. A Case for the Accountable Cloud. In *LADIS*.

Li, A., Yang, X., Kandula, S. and Zhang, M., 2010. CloudCmp: Shopping for a Cloud Made Easy. In *HotCloud'10*.

Best, R., 1980. Preventing Software Piracy with Crypto-Microprocessors. In *COMPCON 80*.

Dyer, J., Lindemann, M., Perez, R., Sailer, R., Smith, S., Doorn, L. and Weingart, S., 2001. Building the IBM 4758 secure coprocessor. *IEEE Computer*.

Arnold, T., Van Doorn, L., 2004. The IBM PCIXCC: A new cryptographic coprocessor for the IBM eServer. *IBM Journal of Research and Development*, 48(3), pp.475.

Gutmann, P., 2000. An Open-source Cryptographic Coprocessor. In *the 9th USENIX Security Symposium*.