An effective privacy protection scheme for cloud computing

Kuan-Chieh Huang
National Cheng Kung University
13 PUBLICATIONS **119** CITATIONS
SEE PROFILE

Yau-Hwang Kuo
National Cheng Kung University
159 PUBLICATIONS **1,720** CITATIONS
SEE PROFILE

An Effective Privacy Protection Scheme for Cloud Computing

I-Hsun Chuang, Syuan-Hao Li, Kuan-Chieh Huang, Yau-Hwang Kuo

Center for Research of E-life DIgital Technology (CREDIT), Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan

{jacky, shayneli, kchuang, kuoyh}@ismp.csie.ncku.edu.tw

Abstract—**With the rapid development of cloud computing, more cloud services are into our daily life, and thus security protection of cloud services, especially data privacy protection, becomes more important. However to perform privacy protection causes huge overhead. Thus it is a critical issue to perform the most suitable protection to decline performance consumption while provide privacy protection. In this paper, the Effective Privacy Protection Scheme (EPPS) is proposed to provide the appropriate privacy protection which is satisfying the user-demand privacy requirement and maintaining system performance simultaneously. At first, we analyze the privacy level users require and quantify security degree and performance of encryption algorithms. Then, an appropriate security composition is derived by the results of analysis and quantified data. Finally, the simulation results show that the EPPS not only fulfills the user-demand privacy but also maintains the cloud system performance in different cloud environments. The execution result of EPPS outperforms other security schemes by 35% to 50%.**

Keywords—**cloud computing, privacy requirement, privacy analysis, security quantification, security composition**

I. Introduction

Cloud computing is an emerging computing style which provides dynamic services, scalable and pay-per-use. The different between cloud computing and other computing models are service-driven, sharing resource, and data hosting in outsourcing storage [1]. Sharing resource makes the hardware performance be used more efficient and provides economic benefits for users to reduce the capital cost and additional expenditure [2]. Data hosting in outsourcing storage lets cloud environment rapidly deliver service to users, and do not spend the waiting time of data transmission required by the services. According to the advantages of cloud computing, we can enjoy more convenient services in our daily life.

However, new security issues are raised at the same time due to the fickle system environment. Different with other computing models, there are no explicit users boundaries or perimeters in cloud computing. The infrastructure is shared to multi-tenants, and users' data are stored and processed in the sharing resource. Since the infrastructure of the sharing resource stored and processed users' data that do not owned by them, users' data may be revealed or breached by other malicious user in the cloud. For this reason, the requirement of data privacy protection in cloud environment becomes more prominent than in other computing models. Nevertheless, some existed data protection mechanisms are invalid because the exact data location in the cloud is uncertain, data may migrate to different servers according to performance or scalability needs. Therefore, providing the solution of data protection is an important issue in the cloud.

A feasible solution for data protection is data encryption. Encryption algorithm offers the benefit of minimum reliance on cloud provider [3]. Thus, users' data can migrate from one provider to another provider without limiting to the specific provider. Furthermore, encryption algorithm protects data no matter where is its physical location. Unfortunately, when performing the encryption algorithm, it often consumes a lot of system resources, such as CPU utilization, and stronger algorithm that generates more significant impact to the system performance. The tradeoff between security and system performance become an important issue when applying an encryption algorithm in cloud environment. In order to provide the data security and maintain cloud system performance, Effective Privacy Protection Scheme (EPPS) in the paper is proposed to resolve the problem. According to user-demand privacy requirement and the result of analyzed related information, the EPPS selects an appropriate security composition that provides enough privacy protection and reduces the extra performance overhead at the same time.

The rest of the paper is organized as follows: Section II introduces the related work. The system architecture and the main concept of EPPS are explained in Section III. In Section IV, the simulation environment and results are display, and the conclusion is described in Section V.

II. Related Work

A. *Cloud Service Models*

At present, it is existed many different types of cloud service models, and three common services models are described as following:

- Software as a Service (SaaS): The applications running on a cloud infrastructure provide service to consumer, and it is accessible from various clients through a thin client interface such as a web browser. Some examples are Google Apps (mail, docs, and etc.) and Salesforce.

- Platform as a Service (PaaS): Service provider provides a specific cloud environment, some software tools and programming language to consumer for developing, testing, and hosting their applications. In this service model, the consumer does not control or manage the underlying cloud infrastructure. An example of PaaS is Google App engine.
- Infrastructure as a Service (IaaS): IaaS allows consumer to rent hardware include processors, storages, network, and other fundamental computing resources. In this service model, consumers do not control or manage the underlying cloud infrastructure directly. They control the computing resources through operating systems.

B. *Amazon Web Services*

Amazon is an e-commerce company which selling electric business in the beginning, and now becomes the famous cloud services provider that provides web services in IaaS model. Amazon Web Services (AWS) composed by a set of remote computing services offers computing power and storage for users to develop their applications or software [4]. The most central and notable services of AWS are Elastic Compute Cloud (EC2) and Simple Storage Service (S3).

In security aspect, AWS provides the protection for network, Virtual Machine (VM), and physical datacenter [5]. Some network security issues like attacks of Distributed Denial of Service, Man in the Middle, and IP Spoofing have been protected by AWS. Furthermore, the security of VM in AWS is protected via Xen hypervisor that ensures the isolation of each instance. The security of physical datacenter is implemented by using video surveillance, intrusion detection system, and authorized staff to authenticate the person who can access datacenter. AWS provides almost completely protection in authentication and non-repudiation, but the confidentiality of users' data does not be considered. Thus, the data processing and storing in AWS is unsafe, and it becomes an important issue that offers a strong privacy protection for the confidentiality of users' data.

In recent years, some data protection mechanisms are proposed to resolve the above problem. [6, 7] apply encryption mechanisms to provide the data confidentiality protection. However, they do not take into account the system performance impact for their protection mechanism, when their methods are implemented that may cause significant performance overhead. [8] discusses the influence of system performance for data encryption and proposes an approach for tradeoff between confidentiality and performance. Nevertheless, it makes users' data dangerous in sometimes that disclose its content. No one proposes a mechanism that protects data and considers the performance overhead at the same time. Thus, we want to propose a scheme that secure users' data confidentiality in the cloud storage and maintain cloud system performance without much extra system overhead.

C. *Cracking Year*

Security is an abstract concept, and how to quantify the strength of different encryption algorithms and mapping to a value becoming an important issue. The Cracking Year is an objective way to evaluate the strength of an encryption algorithm, since it represents the complexity of cracking the encryption algorithms. [9] proposes a method to derive the key-size that shows in (1) for encryption algorithms based on mathematical cryptanalysis and an estimate of available resources. The infeasible key size of a specific year y is derived in the $IKS(y)$ function. If the $IKS(y)$ is greater than real key size which used by the algorithm, y is the year which the algorithm will be cracked.

$$IKS(y) = 56 + (y - DESTrust) \times \left(\frac{12}{TechProgress} + \frac{BudgetDepend}{Budget} \right) - \log_2(SymCphrPerf) \qquad (1)$$

where $SymCphrPerf$ is the ratio of number of cycles for using DES to encrypt a single block to number of cycles for using other symmetric key cipher to encrypt a single block.. Default values of *DESTrust* and *TechProgress* are 1982 and 18, and *Budget* and *BudgetDepend* are 10 and 1.

III. Effective Privacy Protection Scheme

In this section, we describe a Cloud Data Protection System (CDPS) that includes the detailed EPPS and its main concepts.

A. *System Architecture*

Figure 1 shows the architecture of CDPS, the selecting protection mechanism in the top half determines a composition of encryption algorithm and the division numbers to protect users' data. The bottom half is data protection flow that data will be protected by implementing system selecting security composition. The system contains four major components – Privacy Analysis, Quantification Models, Data Division, and Data Protection Procedure.

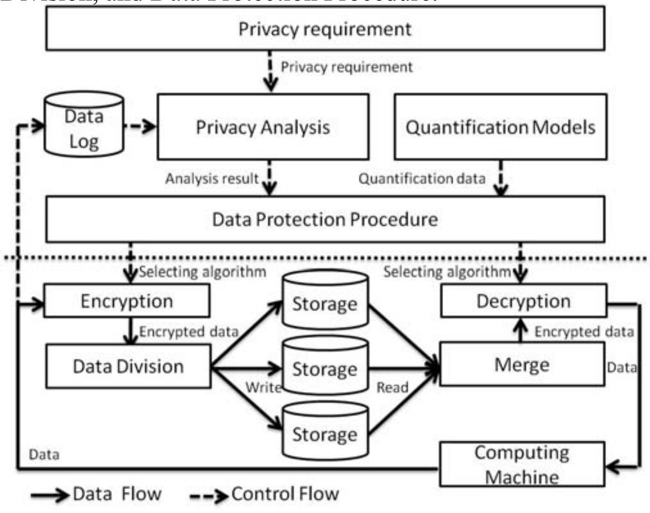

Figure 1. CDPS architecture

The privacy analysis in Figure 1 analyzes user-demand privacy requirement and collects the update frequency of key which is used to encrypt data. The quantification models are including the security and speed aspects. The security quantification measures the cracking year of each encryption algorithm used by CDPS, and the speed quantification measures the mega clock cycles per megabyte when executing each encryption algorithm in specific machine. The Data Division is a concept that using to make data more secure. The analysis result and quantification data are used by Data Protection Procedure. The Data Protection Procedure is the kernel function of CDPS, and its major goal is obtaining the composition of encryption algorithm and number of division with maximizing performance in satisfied users' privacy requirement.

B. Privacy Requirement and Analysis

There is no uniform data type when data stored in the cloud. Data stored in the cloud storage has many various data types, like email, video, image, and etc. Each data type has different importance degree for the user in the cloud, because it includes different numbers of sensitive information. For protecting data confidentiality, a security composition is proposed that consists of an encryption algorithm and the number of data division. This is absolutely out of question. Most important data must be protected by strongest security composition. However, if we used the same strong security composition to secure data, they would affect the quality of cloud services when user requires the unimportant data for the service. On the contrary, if the weak encryption was used to provide the protection, it would make user's important data insecure and can be revealed. Hence, we must address the privacy requirements of user's data to do privacy analysis for providing the most appropriate protection.

1) Privacy Level

In order to pick out a suitable security composition, the privacy level is defined to map users' privacy requirement. Users can configure the privacy level according to their data involving how much sensitive information and the security degree they want.

In the paper, the privacy level is divided into three levels, since we believe that users can not clearly distinguish between their privacy requirements more than three levels. In our scenario, the levels can be seen as the kinds of speed, hybrid, and security. They are explained as follows.

- Privacy Level 1 (Speed): The requirement of this level presents that no sensitive information in the data. Users want to use the weak encryption composition to obtain more performance for using cloud services.
- Privacy Level 2 (Hybrid): The requirement of this level presents that data include some sensitive information. If the data uses the weak encryption for protection, users will worry about that the sensitive information is easy to disclose. Nevertheless, users also want to the performance of requiring cloud services not influence too much.
- Privacy Level 3 (Security): In this privacy level, the data contains most important information. In order to protect the data security, users prefer to sacrifice more performance to ensure the confidentiality.

2) Key Update Frequency

After selecting a specific privacy level, the range of security required by users is determined by the privacy level. Before calculating the value of security range required by users, the range of each security $Security_{range}$ is calculated by (2).

$$Security_{range} = \frac{Security_{max}}{\|P_{level}\|} \tag{2}$$

The maximum security $Security_{max}$ is the security score that CDPS can provide by using the most strong encryption algorithm, and its value is 100. The $\|P_{level}\|$ is the number of privacy levels we predefined.

In privacy analysis component, another factor – key update frequency which affects performance and security is also considered. If data in the cloud is written frequently, serious performance overheads will be occurred in strong encryption algorithms. In order to solve this problem, the encryption algorithm is revised according to the key update frequency. The update frequency of key represents the life cycle of a key. If the encryption key is updated frequently, the average of key life cycle is shorter. For this reason, a weaker encryption algorithm is used to secure user's data, since the attackers must cracks the key before it updated or they would have to re-crack again.

The log of data writing is recorded to calculate the times of data written during a recent week, and the update frequency of key is observed by (3).

$$Frequency_{KeyUpdate} = \frac{Write_{data}}{\text{A period } \Delta t} \tag{3}$$

When the data is written frequently, the life cycle of key will relatively reduce. Thus the high performance security algorithm will be selected for better I/O performance.

C. Quantification Models

1) Security Quantification

In CDPS, the security strength of an encryption algorithm is quantified by its cracking year that calculates by (1). Then, the cracking year of each encryption algorithm is normalized to map the security strength into the range between 0 and 100.

2) Speed Quantification

When an encryption algorithm is performed, CPU consumption can be used to evaluate the system performance of encryption. Crypto++ is a security simulation tool that evaluates CPU consumption of encryption algorithms, and it is used to calculate the CPU consumption of all encryption algorithms in CDPS.

D. Data Division

Some of cloud applications are distributed storing the same data in different storages to make the execution more effective in speed aspect, such as MapReduce. It gives us the point to think that how we using the concept of distributed storage in security aspect. Thus, we consider that implementing the data division after encrypting the data. Although it is assumed that data will be able to obtain by attacker in cryptography, the advantage of this method is making data more secure, because the data is encrypted to ciphertext and divided into many parts, and the data can be decrypted only by collecting all of division parts. If attackers cannot take any of all division parts by hacked the storing servers, they cannot recover the encrypted data to crack.

Nevertheless, this method is still depending on the degree of users' confidence. The confidence means the users' trust to believe that provider cannot disclose their data to attackers. It is similar to the secret sharing, but the secret sharing is not suitable used in here due to the cost of space that also called as redundancy is too big. We assume that the hacked probability of every storing division server is the same, and the probability of server hacked is H that the value is between 0 and 1. The probability of all of n storing division servers simultaneously hacked is H^n. If the data is divided into $n+1$ division parts and comparing with n division parts, the probability of collecting all division parts is shown as follow.

$$H^n > H^{n+1} \tag{4}$$

We can find that the probability of all division parts hacked is smaller when the number of division part increasing, and the attacker obtaining all of division parts for recovering the encrypted data is more difficult. In other words, more division parts make data more safety. Therefore, we ensure that using this method in cloud environment can enhance data security, and it can combine with encryption algorithm to offer a security composition for protecting confidentiality. How to decide the composition of encryption algorithm and the number of division part will introduce in following section.

E. Data Protection Procedure

The major goal of Data Protection Procedure is obtaining the optimal composition of encryption algorithm and number of division by objective function and constraint. The procedure is divided into three phases – preparation, selection scheme, and data processing.

1) Preparation

In this phase, CDPS gathers the required parameters of objective function and constraint including the result of analysis from privacy analysis component and the quantification data from quantification models.

2) Selection Scheme

Our selecting principle is finding the composition for expected maximum performance. In other words, the composition will have the minimal delay time t_{delay}. In order to calculate the delay time cost by our scheme, we design following objective function that considering all of the affect factors involving encryption delay and network transmission delay, and it is shown in (5). Because the delay time of encryption depends on encrypted data size B_i and available computing power R_{CPU}, the large encrypted data size or poor computing power will increase the delay. The delay time of network transmission is depended on transmitted data size and available network transmission rate $TR_{network}$, and data division allows us to keep one part division in local.

$$\arg\min t_{delay} = \frac{B_i \times C_j}{R_{CPU}} + \frac{(N-1)/N \times B_i}{TR_{network}} \tag{5}$$

Although the goal is obtaining the composition with minimal delay time, the user-demand privacy requirement must be satisfied at the same time, since we cannot scarify the security of data. Thus, the constraint of the security value of data is calculated by following constrained function shown in (6), and the security value of our selecting composition must be equal or bigger than the lowest security value required by user. Because the data division has not an actual value of security quantification, we regard it as an additional security bonus to multiply the original security provided by encryption algorithm. However, the additional multiply is still depending on the user confidence α. The key update frequency $F_{KeyUpdate}$ is discussing with the encryption security, since it is in time aspect that weak encryption algorithm can provide equivalent security to strong encryption algorithm when the key update period is shorter.

$$\begin{aligned} Security_{data} &= (1+\alpha\times(1-H^n_{data}))\times(S_j \times F_{KeyUpdate}) \\ &\geq Security_{requirement} \end{aligned} \tag{6}$$

3) Data Processing

The state of data is checked in this phase first. If the data is plaintext meaning that it is required to encrypt for protect its confidentiality, a Data Encryption Key (DEK) will random generate and the data is encrypted by the DEK. On the other hand, the data of ciphertext means that it wants to decrypt for using by cloud services.

IV. SIMULATION

In this section, we describe the simulation environment and discuss the simulation results obtained from CDPS in various scenarios.

A. Simulation Environments

In order to similar the architecture of cloud computing IaaS as Amazon EC2, the simulation machine is a PC with Intel Core 2 Duo E8400 3.0GHz CPU, and the operating system is CentOS 5.2 that an open source Linux including the kernel of Xen hypervisor [10]. Because of the Xen virtualization, the machine can enable multiple virtual machines with allocating different physical resource at the same time to fit real cloud environment. The simulation platform is an instance that allocated one core of CPU and installed Windows XP operating system. The related modules in CDPS are implemented by C++ program language to display the advantage of our proposed scheme.

In our simulation, the composition of encryption algorithm with key length and the number of data division is shown in TABLE 1. The encryption algorithms of these compositions conform to the assumption that the stronger encryption algorithm will cause more performance overhead. The number of data division is depended on the encryption algorithm, since the data required stronger algorithm means that it requires more security, and it also meets above assumption. Thus, the number of data division is increasing with the encryption algorithm stronger.

TABLE 1. THE SECURITY COMPOSITIONS OF CDPS

Encryption algorithm	Key length	Number of data division
RC4	104	2
AES	128	3
AES	192	4
AES	256	5
Blowfish	448	6

Furthermore, the default values of the rest parameters used in the function of our proposed selection scheme are as follows. The network transmission rate and data size are 1GBit and 100MB, and the probability of data hacked and user confidence are set to 0.1.

B. Simulation results

1) Performance Enhancement

Our scheme is compared with other security scheme which its encryption algorithm used in our system to measure the average cost time when processing user's random privacy requirement 10000 times. The simulation result is shown in Figure 2. In this simulation, the penalty time is added when the protection of encryption algorithm is failure. If an encryption algorithm cannot achieve the user-demand privacy level in a time, then it is called failure and must be appended the time of penalty for encrypting the data by strongest encryption algorithm again.

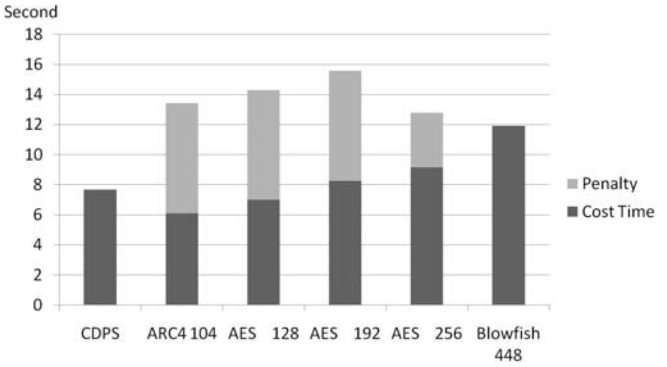

Figure 2. The average cost time

The Cost Time shown in Figure 2 is the original average cost time by using the encryption algorithm, and the Penalty is the average penalty time when using the algorithm failed. We see that the original cost time of our scheme is bigger than in front of three encryption algorithms, but they all have the failure time and must be added the penalty time to achieve the user's requirement. Therefore, their total cost times are all bigger than other encryption algorithms.

We summarize the enhanced performance of CDPS in TABLE 2. The enhancements comparing with other encryption algorithms are up to 50% and at least 35%.

TABLE 2. THE IMPROVEMENT OF CDPS

Compare with	Enhanced Rate (%)
RC4 104	42.80055
AES 128	46.2943
AES 192	50.63245
AES 256	39.92908
Blowfish 448	35.49235

On the other hand, we also analyze the security provided by CDPS and other encryption algorithms to discuss how much security is sacrificed by our scheme. The analyzed result is shown in Figure 3, and the security is represented by using security score calculated by the cracking year. In this analysis, the security score of each encryption algorithm is including two scores, original security score and additional security score. The original security score is obtained by the cracking year of each encryption algorithm. The additional security score is from the strongest encryption algorithm, since the strongest algorithm is used forcibly when an encryption algorithm cannot achieve the user-demand privacy level. In this time, the security score must calculate the cracking year of the strongest encryption algorithm and not the encryption algorithm.

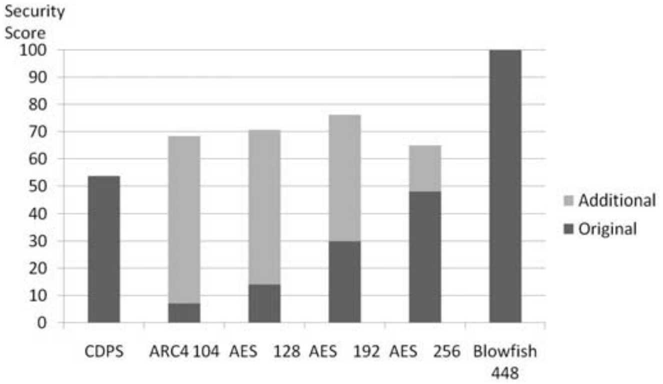

Figure 3. The average security score

In Figure 3, although our scheme sacrifices some security that the total security scores are lower than other encryption algorithms, the original security scores of our scheme are much higher than in front of three encryption algorithms and higher than AES algorithm with 256-bits key size. It means that our scheme can provide better security when comparing with only using one of in front of four encryption algorithms. Because the common used symmetric cryptosystem is AES with 256-bits key size, the security provided by our scheme is still better than it despite we sacrifice some security. Thus, it indicates that our scheme can provide enough security for user to protect their data.

TABLE 3 summarizes the sacrificed security of CDPS comparing with other encryption algorithms, and the worst security loss is 46%. Although the minimum security loss is still 17%, the relatively enhanced performance in TABLE 2 is 39%.

TABLE 3. THE SACRIFICED SECURITY OF CDPS

Compare with	Security Loss (%)
RC4 104	21.43013
AES 128	24.08491
AES 192	29.52761
AES 256	17.50623
Blowfish 448	46.3543

2) Dynamic Computing Resource

When users employ our system, they may request other cloud services at the same time. In cloud environment, the computing power is shared with the other services. Thus, the affect of computing power for our system becomes worthy issues. Because of the feature of Xen hypervisor, the credit schedule [10] limits the computing power for an instance, and the above situation can be simulated by using Xen credit schedule to display the consumed clock cycles of encryption algorithms in different available computing power.

We also compare our scheme with other security schemes to observe the change of cost time in different available computing power, and the result is shown in Figure 4. The result show that the cost time of CDPS is better than others, and the effect is more obvious in the available computing power lower.

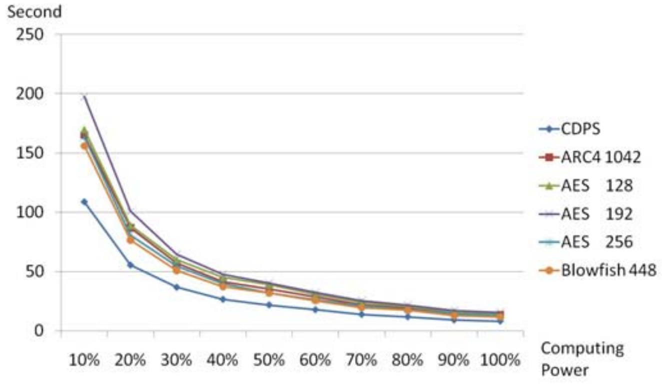

Figure 4. The cost time in different computing power

According to above result, when the available computing power is low, using our scheme can effectively reduce the extra added performance overhead.

V. CONCLUSIONS

In this paper, we propose an Effective Privacy Protection Scheme in cloud environment to secure the confidentiality of users' data without increasing system performance overhead too much. According to different privacy level, our scheme can analyze the related information, selecting the most suitable composition of encryption algorithm and number of data division to provide more secure protection or reduce performance overhead. Finally, the simulation results show that the proposed scheme satisfies user-demand privacy requirement and offers the better performance at the same time. The improved performances comparing with other security schemes are up to 50% and at least 35%.

In the next step, we will analyze more factors, such as different service types. According to the analysis result, CDPS can divide each privacy level into many smaller levels to select more suitable security composition for real user-demand security. By adopting an accurate security analysis, CDPS is capable of improving performance much.

ACKNOWLEDGMENT

The authors would like to thank the National Science Council in Taiwan R.O.C for supporting this research, which is part of the project numbered NSC98-2221-E-006-222-MY3.

REFERENCES

[1] H. Ji and A. Klein, "A Benchmark of Transparent Data Encryption for Migration of Web Applications in the Cloud," in Dependable, Autonomic and Secure Computing, 2009. DASC '09. Eighth IEEE International Conference on, 2009, pp. 735-740.

[2] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono, "On Technical Security Issues in Cloud Computing," in Cloud Computing, 2009. CLOUD '09. IEEE International Conference on, 2009, pp. 109-116.

[3] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing V2.1", Dec. 2009

[4] Amazon Web Services (AWS), Available: http://aws.amazon.com/.

[5] Amazon Web Services: Overview of Security Processes, Available:http://awsmedia.s3.amazonaws.com/pdf/AWS_Security_Whitepaper.pdf.

[6] W. Itani, A. Kayssi, and A. Chehab, "Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures," in Dependable, Autonomic and Secure Computing, 2009. DASC '09. Eighth IEEE International Conference on, 2009, pp. 711-716.

[7] V. D. Cunsolo, S. Distefano, A. Puliafito, and M. Scarpa, "Achieving Information Security in Network Computing Systems," in Dependable, Autonomic and Secure Computing, 2009. DASC '09. Eighth IEEE International Conference on, 2009, pp. 71-77.

[8] R. Prabhakar, S. Seung Woo, C. Patrick, S. H. K. Narayanan, and M. Kandemir, "Securing Disk-Resident Data through Application Level Encryption," in Security in Storage Workshop, 2007. SISW '07. Fourth International IEEE, 2007, pp. 46-57.

[9] Arjen K. Lenstra and Eric R. Verheul, "Selecting Cryptographic Key Sizes," Journal of Cryptology, vol. 14, pp. 255-293, 1999

[10] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. "Xen and the art of virtualization," In Proceedings of the Symposium on Operating Systems Principles (SOSP), Oct. 2003.