Modeling and Improving Security of a Local Disk System for Write-Intensive Workloads

MAIS NIJIM and XIAO QIN
New Mexico Institute of Mining and Technology
and
TAO XIE
San Diego State University, CA

Since security is of critical importance for modern storage systems, it is imperative to protect stored data from being tampered with or disclosed. Although an increasing number of secure storage systems have been developed, there is no way to dynamically choose security services to meet disk requests' flexible security requirements. Furthermore, existing security techniques for disk systems are not suitable to guarantee desired response times of disk requests. We remedy this situation by proposing an adaptive strategy (referred to as AWARDS) that can judiciously select the most appropriate security service for each write request, while endeavoring to guarantee the desired response times of all disk requests. To prove the efficiency of the proposed approach, we build an analytical model to measure the probability that a disk request is completed before its desired response time. The model also can be used to derive the expected value of disk requests' security levels. Empirical results based on synthetic workloads as well as real I/O-intensive applications show that AWARDS significantly improves overall performance over an existing scheme by up to 358.9% (with an average of 213.4%).

Categories and Subject Descriptors: D.4.8 [**Operating Systems**]: Performance—*Simulation, queueing theory*; D.4.6 [**Operating Systems**]: Security a Protection—*Cryptographic controls information flow controls*

General Terms: Security, Performance, Theory

Additional Key Words and Phrases: Quality of security, desired response time, data-intensive applications, security level, local disk

1. INTRODUCTION

In the past decade, storage systems have been an object of substantial interest because of an increasing number of emerging data-intensive

The work was supported in part by the New Mexico Institute of Mining and Technology under Grant 103295 and by Intel Corporation under Grant 2005-04-070.
Author's address: M. Nijim, X. Qin, New Mexico Institute of Mining and Technology, 801 Leroy Place, Socorro, NM 87801; email: {mais,xqin}@cs.nmt.edu; T. Xie, Department of Computer Science, San Diego State University, San Diego, CA 92182.

applications like video surveillance [Avitzour 2004], long running simulations [Tanaka 1993], digital libraries [Sumner and Marlino 2004], remote-sensing database systems [Chang et al. 1997], and out-of-core applications [Qin et al. 2005]. This trend can be attributed to advances in computational power, disk performance, and high-speed networks. There are many cases where data-intensive applications require enriched security to protect data in storage systems from talented intruders [Xie and Qin 2005]. Further, a large number of data-intensive applications require guaranteed response times for interactive or high-priority data [Dimitrijevic and Rangaswami 2003]. Therefore, storage systems are required to provide strong security and guaranteed response times for disk requests. This demanding requirement is increasingly becoming a critical and challenging issue in the development of next-generation storage systems.

Although conventional storage systems are aimed at improving access times and storage space, many existing storage systems are vulnerable to a wide variety of potential threats. As such, existing disk systems fail to meet the security requirements of modern data-intensive applications. To protect storage systems against all possible security threats, researchers have developed various ways of ensuring the security of data in storage systems.

In this article, we seek to present a novel adaptive write strategy for local disk systems, providing a diversity of security services with various qualities of security. The strategy can be seamlessly integrated into disk scheduling mechanisms in disk systems. The proposed strategy is conducive to the achievement of high security for local disk systems, while making the best effort to guarantee desired response times of requests. To prove the efficiency of the proposed approach, we build an analytical model to measure the expected value of security levels and the probability that a disk request is completed before its desired response time.

The rest of the article is organized as follows. In the next section we summarize related work. Section 3 describes the model of disk requests and the new architecture of storage systems. In Section 4, we propose the adaptive write strategy for security-aware storage systems. We build an analytical model in Section 5. Sections 6 and 7 present experimental results based on both synthetic benchmarks (read/write) and real I/O-intensive applications. Finally, Section 8 concludes the article with future directions.

2. RELATED WORK

There is a large body of work on improving performance of disks because disk I/O has become a serious performance bottleneck of computer systems. Previous techniques supporting high-performance storage systems include disk striping [Bordawekar et al. 1994; Salem and Garcia-Molina 1986; Scheuermann et al. 1998], parallel file systems [Cho et al. 1997; Ligon and Ross 1996; Preslan et al. 1999], and load balancing [Qin et al. 2005; Scheuermann et al. 1998],

as well as caching and buffering [Forney et al. 2001; Huber et al. 1995; Ma et al. 2002].

Disk scheduling algorithms also play an important role in reducing the performance gap between processors and disk I/O [Coffman and Hofri 1990; Jacobson and Wilkes 1991; Seltzer et al. 1990; Yu et al. 1993]. The shortest-seek-time-first (SSTF) algorithm is efficient in minimizing seek times, but starvation-bound and unfair in nature [Denning 1967]. The SCAN scheduling algorithm can solve the unfairness problem while optimizing seek times [Denning 1967]. Reist and Daniel proposed a parameterized generalization of the SSTF and SCAN algorithms [Reist and Daniel 1987]. However, the aforementioned disk scheduling algorithms are unable to guarantee desired response times of disk requests.

Many data-intensive applications require that data is stored or retrieved before a desired response time [Reuther and Pohlack 2003]. The SCAN-EDF can be employed to fulfill this requirement [Seltzer et al. 1990]. Recently, many disk schedulers were implemented for a mixed-media dataset, a mixture of data accessed by multimedia applications and best-effort applications [Balafoutis et al. 2003; Bosch and Mullender 2000]. Several disk scheduling algorithms were proposed to provide quality of service guarantees to different classes of applications [Bruno et al. 1999; Reddy and Wyllie 1999; Shenoy and Vin 1998]. The salient difference between the proposed approach and existing disk scheduling algorithms in the literature is that our strategy is focused on maximizing security of a local disk. Moreover, our strategy is orthogonal to existing disk scheduling policies in the sense that the novel strategy can be readily integrated into existing disk schedulers to improve security of local disks.

In recent years, the issue of security in storage systems has been addressed and reported in the literature. Riedel et al. developed a common framework of core functions required for any secure storage system [Riedel et al. 2002]. To protect data in untrusted storage systems, researchers designed and implemented cryptographic file systems where data is stored in encrypted form [Blaze 1993; Hughes and Corcorna 1999]. Several key distribution schemes were proposed in SFS [Mazieres et al. 1999] and SNAD systems [Miller et al. 2002]. Although a variety of secure storage systems were implemented, there is no adaptive way of choosing security services to meet disk requests' flexible security requirements. Furthermore, the preceding security techniques are not suitable for disk requests with desired response times. We remedy this situation by proposing an adaptive strategy that can judiciously choose the most appropriate security service for each write request, while making the best effort to guarantee the desired response times of all disk requests.

In our previous work, we proposed a family of dynamic security-aware scheduling algorithms for clusters [Xie and Qin 2005; Xie et al. 2005] and grids [Xie and Qin 2005]. Unfortunately, these scheduling algorithms limit their applicability to computing resources and thus, our previous algorithms can not be employed to storage systems.

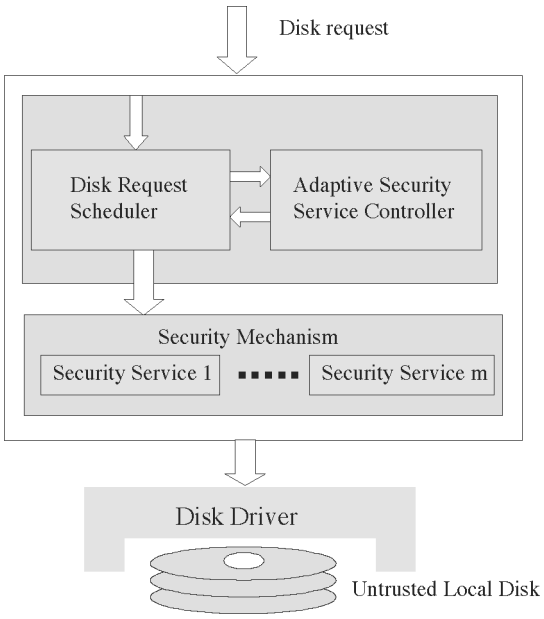

Fig. 1. Architecture of a security-aware storage system.

3. ARCHITECTURE AND DISK REQUESTS WITH SECURITY REQUIREMENTS

3.1 Architecture of a Security-Aware Storage System

In this study we focus on local disk systems, and both storage systems in parallel and distributed environments are out of the scope of this article. Our new algorithm is based on a security-aware storage system architecture. Figure 1 depicts the main components, that is, a disk driver, security mechanism, and disk scheduling core, of this architecture.

The architecture is briefly overviewed as follows. The disk driver is responsible for controlling access to an untrusted local disk. The security mechanism provides an array of security services to guard data blocks residing in the local disk against unauthorized access and information theft. Without loss of generality, we consider in this study only confidentiality security services, where it is assumed that keys are known only to the owner, reader, and writers [Blaze 1993]. The security mechanism can be readily extended to employ integrity and availability services. The disk scheduling core consists of two parts: a disk scheduler and an adaptive security service controller. While the scheduler implements generic logic and timing mechanisms for scheduling and waiting, the security service controller dynamically chooses the most appropriate security service for each disk request. Since the security service controller is independent of disk scheduling policies, the service controller is implemented separately for each disk scheduler. As such, it is easy to apply the security service controller to any disk scheduling policy implementation.

3.2 Modeling Disk Requests With Security Requirements

Each disk request (that is submitted a security-aware storage system specifies quality of service) including security and performance requirements. A security requirement is defined as a lower bound security level, which is a value from 0.1 to 1.0. A performance requirement is posed as a desired response time. The quality of service requirements of disk requests can be derived from applications issuing these disk operations. Security quality of encryption services implemented in the security mechanism are measured by security levels. An encryption service with a high security level means the high quality of security provided by the service. For example, a disk request specifies a lower bound security level as 0.4. In this case, encryption services with security levels higher than or equal to 0.4 can successfully meet the disk request's security requirements.

A disk request r is characterized by five parameters: $r = (o, a, d, s, t)$, where o indicates that the request is a read or write, a is the disk address, d is the data size measured in KBs, s is the lower security-level bound, and t is the desired response time.

The security benefit gained by a disk request r_i can be measured by the security level i of an encryption service facilitating confidentiality for the disk request. Likewise, the quality of security offered by a local storage system can be measured by a sum of security benefits of all incoming disk requests. Let R be a set of incoming disk requests. Our proposed AWARDS strategy strives to maximize the security benefit of the storage system. Thus, we can obtain the following non-linear optimization problem formulation to maximize the security benefit, where i is the real response time of the ith disk request.

$$\begin{aligned} &Maximize \sum_{r_i \in R} \sigma_i \\ &Subject\ to\ \forall r_i \in R : s_i \leq \sigma_i \leq 1, and\ \rho_i \leq t_i \end{aligned} \tag{3.1}$$

3.3 Security Overhead Model

Now we consider security overhead incurred by confidentiality services. The security overhead model can be easily extended to incorporate other security services. Encryption is used to encrypt data blocks residing in local storage systems. There are ten encryption algorithms (see Table 1) implemented in the security mechanism. Based on the encryption algorithms' performance, each algorithm is assigned a security level. For example, level 0.9 implies that we use 3DES, which is the strongest yet slowest encryption function among the alternatives. Note that the overhead of encryption depends on the chosen cryptographic algorithm and the size of data block. Figure 2 plots enciphering time in seconds as a function of the encryption algorithms and data size on a 175 MHz Dec Alpha600 machine [Nahum et al. 1995; Xie et al. 2005; Xie and Qin 2005]. Let be the security level of r_i, and the security overhead can be calculated using Eq. 3.2, where d_i is the data size and *P(i)* is a function used to map a

Table I. Cryptographic Algorithms Used for Encryption Services

Cryptographic Algorithms	Security Level,σ	Performance(KB/ms),P(σ)
SEAL	0.1	168.75
RC4	0.2	96.43
Blowfish	0.3	37.5
Knufu/Khafre	0.4	33.75
RC5	0.5	29.35
Rijndael	0.6	21.09
DES	0.7	15
IDEA	0.8	13.5
3DES	0.9	6.25

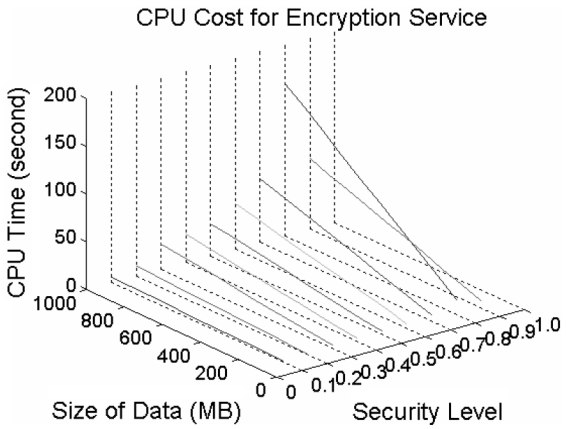

Fig. 2. Security overhead of encryption services.

security level i to the corresponding performance of encryption service listed in Table 1.

$$T_{security}(\sigma_i, d_i) = \frac{d_i}{P(\sigma_i)} \tag{3.2}$$

4. THE ADAPTIVE WRITE STRATEGY

This section presents the proposed adaptive write strategy, which is referred to as AWARDS throughput this article. We assume that the overhead of AWARDS is negligible when compared to the processing times of disk requests. In this study we consider a local disk system providing nine encryption services with different security levels (see Section 3.3). AWARDS aims at improving the quality of security for local disk systems. To achieve this goal, AWARDS aggressively increases the security level of each incoming disk request under the condition that the request's response time does not exceed the desired response time. We make use of an example to elaborate the basic idea behind the AWARDS strategy. Suppose there are three write requests submitted to a local disk system at time 0. Table 2 shows important parameters of the three write requests. We

Table II. Important Parameters of the Three Write Requests

R	d_i	s_i	t_i	T	σ_i
r_1	90KB	0.2	18ms	17.7ms	0.8
r_2	150KB	0.1	41ms	40.7ms	0.7
r_3	30KB	0.3	55ms	54.5ms	0.9

Here, requests(R), data size(d_i), minimal security levels(s_i), desired response time(t_i), response time under AWARDS(T), security level under AWARDS(σ_i).

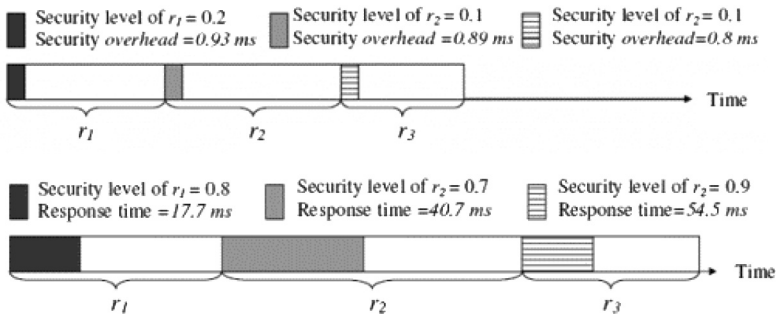

Fig. 3. (a: top) security levels and overhead of the requests. The system is running without AWARDS. (b: bottom) security levels and response times of the requests. The system is running with AWARDS.

assume that the disk bandwidth is 30MB/s, and that for each request, the sum of rotational latency and seek time is 8ms.

Prior to writing the data of a request to the local disk, the disk system enciphers the data using a selected encryption service. Similarly, the system deciphers the data of a read request after the data is retrieved from the disk. Hence, the processing time of each disk request logically consists of two parts: security overhead (indicated by a shaded block) and disk service time (represented by an unshaded block; see Eq. 4.1 and Figure 3). Figure 3(a) illustrates the security levels and response times of the requests when the disk system is running without AWARDS. In this case, the minimal security requirement of each request is met at the minimal security overhead. In contrast, AWARDS can significantly increase the security levels, provided that the desired response times can be guaranteed (see Figure 3(b)). For example, the response times of the three requests are 17.7ms, 40.7ms, and 54.5ms, which are less than the respective desired response times (see Table 2). Specifically, with the AWARDS strategy in place, security levels are improved by an average of 366.7%.

Now we present the AWARDS strategy, which adaptively adjusts the security levels of write requests (see Figure 4). It is worth noting that AWARDS is unable to adjust the security levels of read requests because the local disk system has to use a corresponding encryption service to decipher the data of a read request after the cipher is read from the disk. Before increasing the security level of a write request r_i, AWARDS must ensure that r_i and those write requests with earlier desired response times can be finished before their desired response

1. **Input:** r_i, a disk request r_i submitted to the local disk system; Q, a waiting queue
2. Insert ri into Q based on the earliest desired response time first policy;
3. **for** each write request r_j in the waiting queue Q **do**
4. $\sigma_j \leftarrow s_j$; /* Initialize the security levels */
5. **for** each write request rj in the waiting queue Q **do**
6. **if** $\sigma_j < 0.9$ **then** /* σj can be further increased */ (see **property 1(1)**)
7. Increase security level σj by 0.1;
8. **for** each request r_u with longer desired response time **do** (see **property 1(2)**)
9. **if** r_u can not be completed before its desired response time **then**
10. Decrease security level σ_j by 0.1; break;
11. **end for**
12. **if** σ_j is not increased in Step 7 **then** break;
13. **end for**

Fig. 4. The adaptive write strategy for local disk systems.

times (see condition (4.2) in the following property). Therefore, the following property needs to be satisfied in the AWARDS strategy.

Property 1. If the security level of a write request r_i is increased by 0.1, the following conditions must hold:

$$\text{(1) The current security level of } r_i \text{ is less than 0.9, that is, } \sigma_i < 0.9; \quad \text{and} \tag{4.1}$$

$$\text{(2) } \forall r_k \in Q, t_k \geq t_i : \text{es}(r_k) + T(r_k, \sigma_k) \leq t_k, \tag{4.2}$$

where Q is the waiting queue, $es(r_k)$ is the start time of request r_k, and $T(r_k, \sigma_k)$ is the processing time of $r_i \in Q$. The start time of $es(r_k)$ can be expressed by

$$es(r_k) = \sum_{r_l \in Q, t_l \leq t_l} T(r_l, \sigma_l), \tag{4.3}$$

where the expression on the right side of Eq. (4.3) is the total processing time of disk requests whose desired response times are earlier than that of r_k. The processing time $T(r_k, \sigma_k)$ in condition (4.2) can be computed by

$$T(r_i, \sigma_i) = T_{seek}(a_i) + T_{rot}(a_i) + \frac{d_i}{B_{disk}} + T_{security}(d_i, \sigma_i), \tag{4.4}$$

where T_{seek} and T_{rot} are, respectively, the seek time and rotational latency, $\frac{d_i}{B_{disk}}$ is the data transfer time that largely depends on data size d_i and disk bandwidth B_{disk}, and $T_{security}(\sigma_i, d_i)$ is the security overhead that lies in the security level and security-critical data size (see Eq. (3.2)).

The adaptive write strategy for secure local disk systems, or AWARDS, is described in Figure 4. AWARDS aims at improving the quality of security and guaranteeing disk requests' desired response times. To achieve high security, the AWARDS strategy optimizes the security levels of write requests (see Step 7 in Figure 4).

Upon the arrival of a disk request, AWARDS inserts the request into the waiting queue based on the earliest desired response time first policy, meaning that disk requests with earlier response times are processed first. Before proceeding to the optimization of security levels of write requests in the queue, AWARDS first initializes the security levels of all write requests to minimal

levels (see Step 4). Step 7 then gradually enhances the security level of each request r_j under the conditions that: (1) The current security level of r_j does not exceed 0.9 (see Step 6); and (2) the desired response times of requests being processed later than r_j can be achieved (see Steps 8–10). The process of optimizing security levels repeatedly performs (see Step 5); and stops when a request's desired response time cannot be met (see Step 12). In doing so, the AWARDS strategy can maximize the security levels of write requests (see Step 7), while guaranteeing the desired response times of all disk requests in the queue (see Steps 9 and 10). The time complexity of AWARDS is evaluated as follows.

THEOREM 1. *The time complexity of AWARDS is $O(n^2)$, where n is the number of disk requests in the waiting queue.*

PROOF. To increase the security level of a request, it takes $O(n)$ time check condition (4.2) (see Step 8). Since there are $O(n)$ numbers of write requests in the waiting queue, the time complexity of optimizing the security levels of write requests is: $O(n)O(n) = O(n^2)$.

5. ANALYTICAL MODEL

In this section, we first derive the probability that a disk request is completed before its desired response time. Second, we calculate the expected value of security levels assigned to disk requests with security requirements.

5.1 Satisfied Ratio

We first calculate, for every disk request r_i submitted to a local disk system, the probability that r_i can be finished within the desired response time t_i, that is, $Pr(\rho_i \leq t_i)$, where ρ_i is the real response time. At any time when a disk request arrives, the request is inserted in the queue such that all requests with earlier desired response times will be given higher priority and executed first. Note that n waiting requests in the queue are indexed by their priorities so that the desired response time of r_i is smaller than that of r_j if $i < j$, that is, $\forall 1 \leq i, j \leq n : i < j \Rightarrow t_i < t_j$. In this study we assume that the processing times of different disk requests are statistically independent.

Recall that $T(r_i,\sigma_i)$ is the processing time of a disk request $r_i \in Q$. Moreover, $T(r_i, \sigma_i)$ is computed by Eq . (4.4). Let p_x be the probability that the disk request r_i requires x time units to complete, that is, $p_x = P_r(T(r_i, \sigma_i) = x)$. Similarly, let q_y be the probability that the total required processing time of disk requests with higher priorities is y, that is, $q_y = Pr\left[\sum_{j=1}^{i-1} T(r_j, \sigma_j) = y\right]$. The probability that r_i is unable to be finished within the desired response time t_i is computed as follows:

$$
\begin{aligned}
Pr(\rho_i > t_i) \;=\; & Pr\left[T(r_i,\sigma_i) = 1 \,\middle|\, \sum_{j=1}^{i-1} T(r_j,\sigma_j) \geq t_i\right] + \\
& \vdots \\
& + Pr\left[T(r_i,\sigma_i) = k \,\middle|\, \sum_{j=1}^{i-1} T(r_j,\sigma_j) \geq t_i + 1 - k\right] +
\end{aligned}
$$

$$
\begin{aligned}
&\vdots \\
&+Pr\Bigg(\Bigg[T(r_i,\sigma_i)=t_i+1\Bigg|\sum_{j=1}^{i-1}T(r_j,\sigma_j)\geq 0\Bigg]\\
&=\ p1\sum_{y=t_i}^{\infty}q_y+p2\sum_{y=t_i-1}^{\infty}q_y+\cdots+p_{t_i}\sum_{y=1}^{\infty}q_y+p_{t_i+1}\sum_{y=0}^{\infty}q_y\\
&=\ \sum_{x=1}^{t_i+1}\Bigg[p_x\sum_{y=t_i+1-x}^{\infty}q_y\Bigg], \qquad (5.1)
\end{aligned}
$$

where the second line in the preceding equation indicates the conditional probability that the required processing time of disk request r_i is k, given that it requires at least t_i+1-k time units to complete disk requests with higher priorities.

The probability that a disk request r_i is completed within its desired response time is given by

$$
\begin{aligned}
Pr(\rho_i\leq t_i)\ &=\ 1-Pr(\rho_i>t_i)\\
&=\ 1-\sum_{x=1}^{t_i+1}\Bigg[p_x\sum_{y=t_i+1-x}^{\infty}q_y\Bigg]. \qquad (5.2)
\end{aligned}
$$

5.2 Quality of Security

To evaluate quality of security for a local disk system, we derive in this section the expected security level experienced by disk requests. Before proceeding to the calculation of the expected security level, we compute the probability $Pr(\sigma_i=z)$ that the security level of each submitted disk request r_i equals z. Recall that the security level of a disk request relies on the desired response time, the data size of the request, and processing times of other waiting requests with higher priorities. As such, $Pr(\sigma_i=z)$ can be calculated as

$$
\begin{aligned}
Pr(\sigma_i=z)=Pr(d_i=d_{min}).&\sum_{j=t_{min}}^{t_{max}}\Bigg\{Pr(t_i=j).Pr\Bigg[\sum_{k=1}^{j-1}T(r_k,\sigma_k)=j-\bar{T}(d_{min},z)\Bigg]\Bigg\}\\
&\vdots\\
&+Pr(d_i=l).\sum_{j=t_{min}}^{t_{max}}\Bigg\{Pr(t_i=j).Pr\Bigg[\sum_{k=1}^{i-1}T(r_k,\sigma_k)=j-\bar{T}(l,z)\Bigg]\Bigg\}\\
&\vdots\\
&+Pr(d_i=d_{max}).\sum_{j=t_{min}}^{t_{max}}\Bigg\{Pr(t_i=j).Pr\Bigg[\sum_{k=1}^{i-1}T(r_k,\sigma_k)=j-\bar{T}(d_{max},z)\Bigg]\Bigg\}\\
=\sum_{l=d_{dmin}}^{d_{max}}&\Bigg\{Pr(d_i=l).\sum_{j=t_{min}}^{t_{max}}\Bigg\{Pr(t_i=j).Pr\Bigg[\sum_{k=1}^{i-1}T(r_k,\sigma_k)=j-\bar{T}(l,z)\Bigg]\Bigg\}\Bigg\}, \qquad (5.3)
\end{aligned}
$$

Table III. Disk Parameters

IBM Ultrastar 36Z15	
Size	18.4GB
RPM	15000
Seek Time, T_{seek}	7.18 ms
Rotational Time, T_{rot}	4.02 ms
Disk Bandwidth, B_{disk}	30 MB/s

where $\bar{T}(l, z)$ is the processing time of r_i if the data size is l and the security level is set to z, that is,

$$\bar{T}(l, z) = T_{seek}(a_i) + T_{latency}(a_i) + [d_i/B_{disk}] + T_{security}(l, z).$$

The data size and desired response time of each disk request are two random variables distributed according to two probability density functions, which are known *a priori*. We let u_l denote the probability that the data size of the disk request is l, and let v_j denote the probability that the desired response time equals j. The minimal and maximal data sizes are represented by d_{min} and d_{max}, respectively. Likewise, the minimal and maximal desired response times are denoted by t_{min} and t_{max}. Based on Eq. (4.4), the probability $Pr(\sigma_i = z)$ can be expressed as

$$Pr(\sigma_i = z) = \sum_{l=d_{dmin}}^{d_{max}} \left\{ u_i . \sum_{j=t_{min}}^{t_{max}} \left[v_i . q_{j-\bar{T}(l,z)} \right] \right\}, \tag{5.4}$$

where $q_{i-\bar{T}(l,z)} = Pr(\sum_{k=1}^{i-1} T(r_k, \sigma_k) = j - \bar{T}(l, z))$.

The expected security level experienced by disk requests with security requirements can be directly derived from Eq. (4.5). Thus, the expected security level is given by

$$E(\sigma) = \sum_{i=1}^{9} ([i/10].Pr(\sigma = [i/10])). \tag{5.5}$$

6. SYNTHETIC BENCHMARKS

To quantitatively evaluate the performance of the AWARDS strategy, we implemented a basic prototype of AWARDS to work with a simulated local disk system. Workloads with both synthetic benchmarks and real-world I/O-intensive applications (see Section 7) were generated and evaluated in the prototype of AWARDS.

Our experimental test consisted of the prototype, the simulated disk system, and the nine encryption services described in Table 2. Disk parameters, summarized in Table 3, are similar to those of the IBM Utrastar 36Z15.

The following four important performance metrics are used to evaluate the AWARDS strategy: (1) *Satisfied ratio* is defined as a fraction of the total arrived disk requests that are found to be completed before their desired response times;

Table IV. Workload Configuration

Parameter	Value(Fixed)–(Varied)
Disk Bandwidth	30KB/s
Request Arrival Rate	(0.1,0.2,0.3,0.4,0.5) No./s
Desired Response Time	10s
Security Level	(0.5)–(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9)
Write Ratio	(100%)–(0%, 10%, 20%, 30%, 100%)
Data Size	(500KB)–(300,400,500,600,700)KB

(2) *average security level* is the average value of security levels achieved from all disk requests issued to the disk system; (3) *average security overhead* is measured in seconds; and (4) *overall performance* is measured by a product of the satisfied ratio and average security level.

To provide fair comparisons, we obtained empirical results based on a wide range of synthetically generated workload and environmental conditions, which closely resemble a variety of I/O-intensive applications. Table 4 outlines important workload configuration parameters for the simulated disk system used in our experiments.

In Section 6.1 we present performance comparisons between a disk system with AWARDS and another system not employing AWARDS (referred to as the Original strategy). Section 6.2 studies the impact of data size on disk performance. Section 6.3 examines performance sensitivities to disk bandwidth. The performance impact of security requirements is evaluated in Section 6.4. Finally, Section 6.5 demonstrates the performance impact of an increasing write ratio.

6.1 Overall Performance Comparisons

This experiment is aimed at comparing AWARDS against Original, which is a strategy without making use of AWARDS. To stringently evaluate the performance of AWARDS and its competitive strategy, we set the write ratio to 100%. The impact of write ratio on system performance is studied in Section 6.5. We increased the disk request arrival rate from 0.1 to 0.5 No./s. Other workload parameters were fixed to the same values as those listed in Table 4.

Figure 5 plots the four performance metrics for the AWARDS and Original strategies. Figure 1(a) reveals that AWARDS maintains a very close performance in satisfied ratio to the Original strategy. Figure 5(b) shows that AWARDS significantly outperforms Original in average security level by an average of 138.2%. As the request arrival rate increases, the average security levels of the two strategies decrease. However, AWARDS always achieves higher average security levels compared with those of the Original strategy. This result can be explained in part by Figure 5(c), which shows that the average security overhead of AWARDS is constantly higher than that of the alternative. In other words, the high security levels of AWARDS are achieved at the cost of high security overheads. Figure 5(d) clearly reveals that AWARDS outperforms Original in terms of overall performance. Specifically, AWARDS obtains an improvement in overall performance over the Original strategy by an average of 125.6%. The

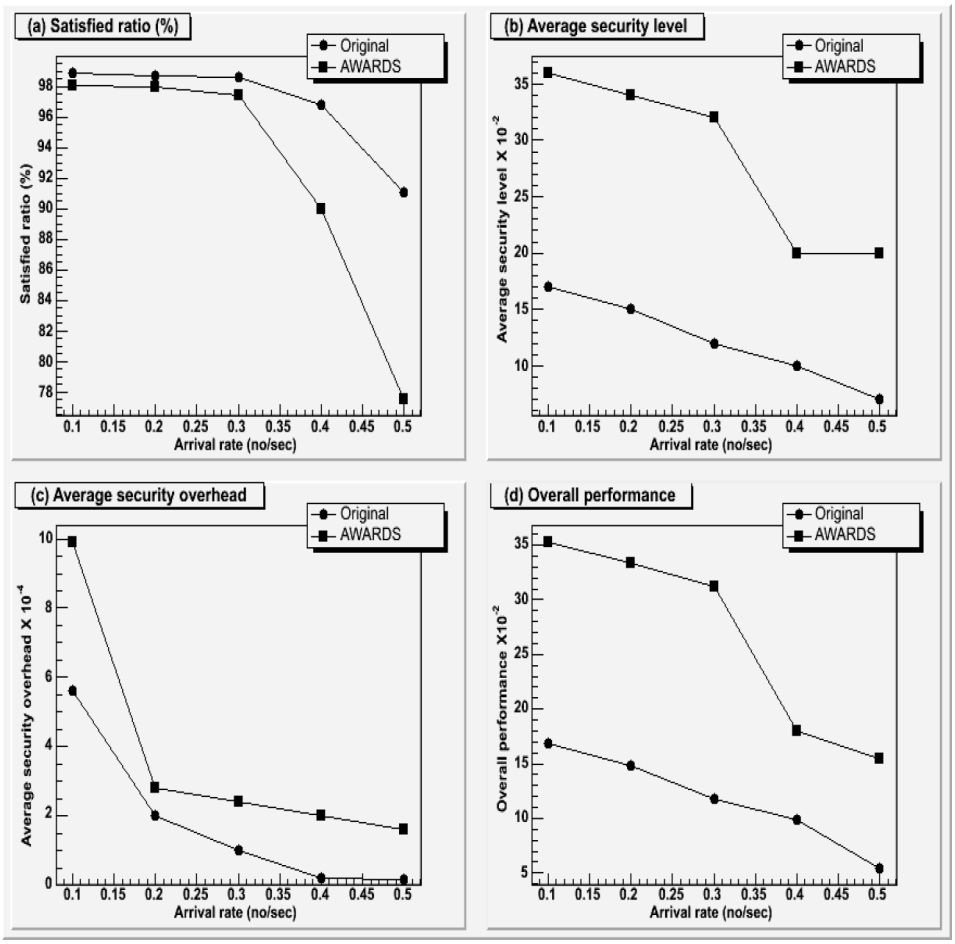

Fig. 5. Performance impact of arrival rate.

performance improvement can be attributed to the fact that AWARDS adaptively enhances the security levels of each write request, under the condition that all requests in the disk can achieve their desired response times.

6.2 Impact of Data Size

In this section, we varied the data size from 300 to 700KB to examine the performance impact of data size on the local disk system. Again, other workload parameters were kept unchanged (see Table 4).

Figure 6(a) shows that when the data size increases from 300 to 700KB, the AWARDS strategy delivers similar satisfied ratios to those of Original. This result, which is consistent with the result presented in Figure 5(a), demonstrates that AWARDS achieves good performance in satisfied ratio. Like Figure 5(b), Figure 6(b) shows a significant improvement of AWARDS in security level over Original.

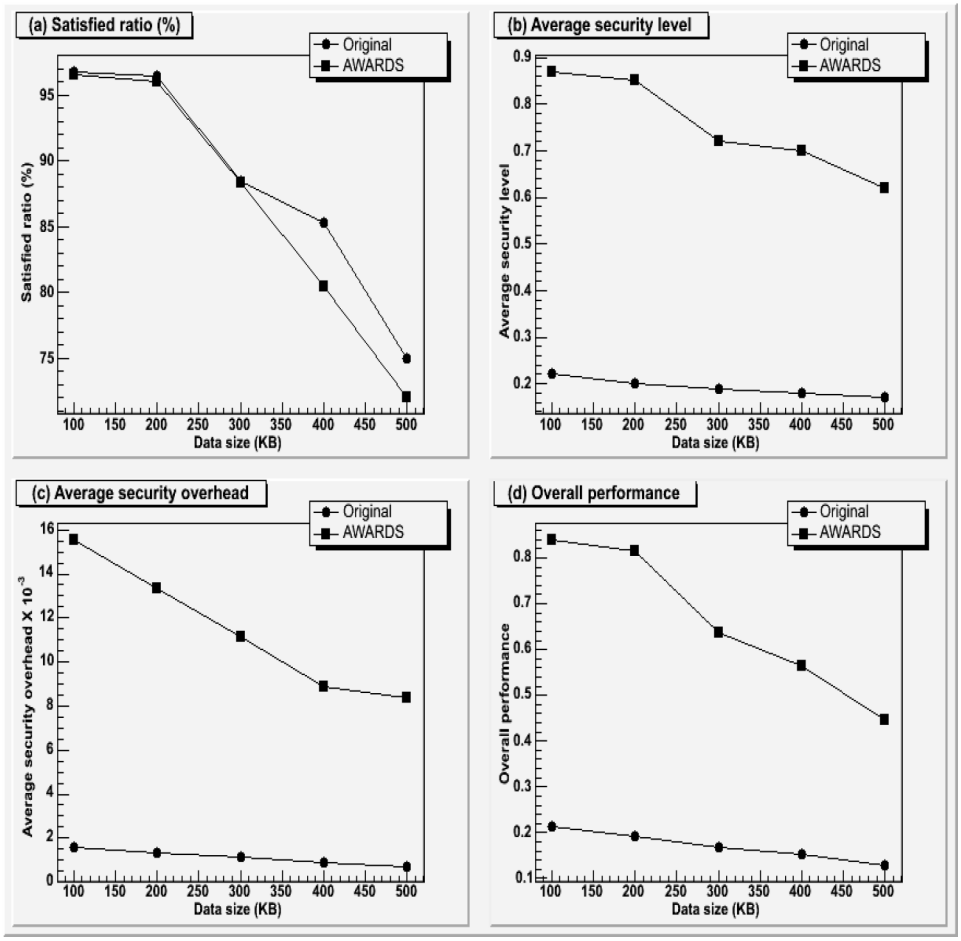

Fig. 6. Performance impact of data size.

Interestingly, it is observed from Figure 6(b) that the average security level gradually drops with increasing value of data size. This is because increasing data size results in an overloaded condition, which in turn leads to decreasing security levels of disk requests in the queue in order to finish most requests before their desired response times. Further, we observe from Figures 6(c) and 6(d) that when the data size goes up, the average security overhead and overall performance of AWARDS decreases, because the security levels of disk requests are lowered due to the high workload. By contrast, the average security level and overhead of the Original strategy only slightly reduce with increasing value of data size. These results indicate that Original is insensitive to data size.

6.3 Impact of Disk bandwidth

In this experiment we investigated the performance of AWARDS and Original when the disk bandwidth varies from 10MB/s to 50MB/s. An important observation drawn from Figure 7(a) is that as bandwidth increases,

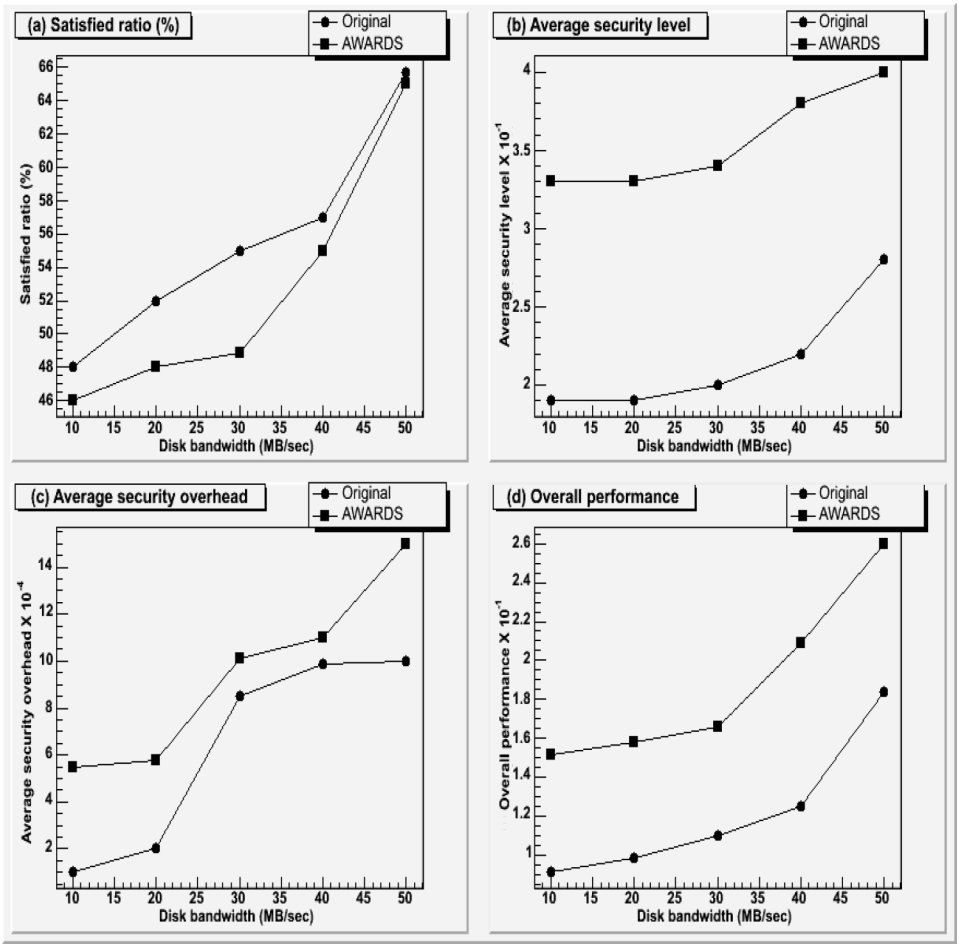

Fig. 7. Performance impact of disk bandwidth.

the satisfied ratios of the two strategies rise gently. The result can be explained by the fact that high disk bandwidth leads to short transfer times, which in turn result in short processing times of disk requests. Consequently, more disk requests can be finished before their respective desired response times.

It is worth noting that the satisfied ratio curves of the two alternatives begin to merge when the bandwidth is larger than 40MB/s. Figure 7(b) shows that the average security level increases as disk bandwidth is increased because the processing times of disk requests become smaller in light of high disk bandwidth. These shortened processing times enable AWARDS to further increase security levels at the expense of higher security overheads (see Figure 7(c)). Thanks to the increasing satisfied ratio and average security level, the overall performance of AWARDS is substantially boosted in case of a disk system that provides high disk bandwidth (see Figure 7(d)).

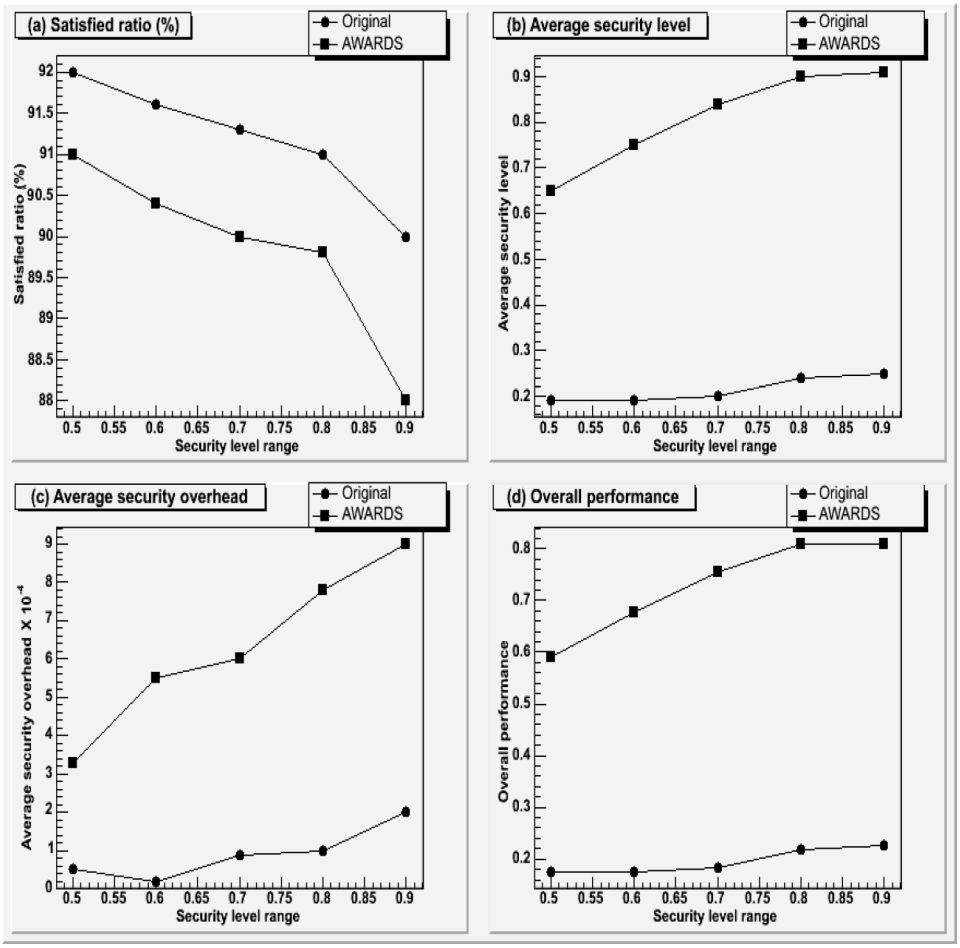

Fig. 8. Performance impact of security level range.

6.4 Security Level Range

In this group of experiments, we studied the performance impact of the security requirements of disk requests. The maximal required security level is varied from 0.5 to 0.9, whereas the minimal required security level is fixed at a value of 0.1.

It is observed from Figure 8(a) that when maximal required security levels increase, the satisfied ratios of the two strategies decrease. The main reason for this result is that when disk requests require higher security levels, the security overhead inevitably grows (see Figure 8(c)). This growing security overhead in turn causes a significant drop in the satisfied ratio. We observe from Figure 8(b) that the amount of improvement in average security level becomes more prominent with increasing value of the maximal required security level. This performance trend can be explained by the fact that the larger the maximal required security level, the more opportunities for

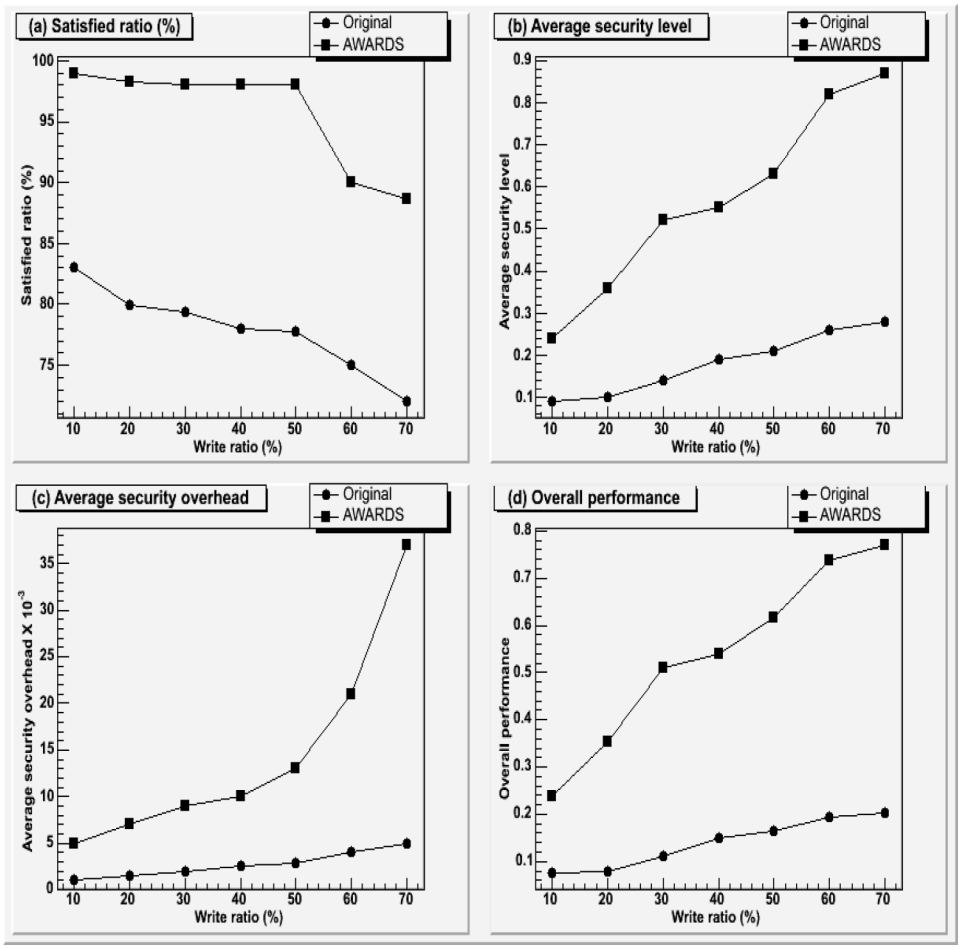

Fig. 9. Performance impact of the write ratio.

AWARDS to dynamically increase the security level of each write request. Because the average security level rapidly increases, the overall performance of AWARDS also rises as the maximal required security levels are increased (see Figure 8(d)).

6.5 Impact of Write Ratio

In the previous experiments, we assume that the write ratio is fixed at 100%. This experiment, however, seeks to measure the impact of the write ratio on disk performance. We increased the write ratio of the workload from 0% to 100% in increments of 10%. Figure 9 plots the four performance metrics as functions of the write ratio.

Like the empirical results presented in Figures 5–8, results shown in Figure 9 illustrates the performance improvements of AWARDS over the alternative. Figure 9(a) shows that the satisfied ratio is marginally reduced

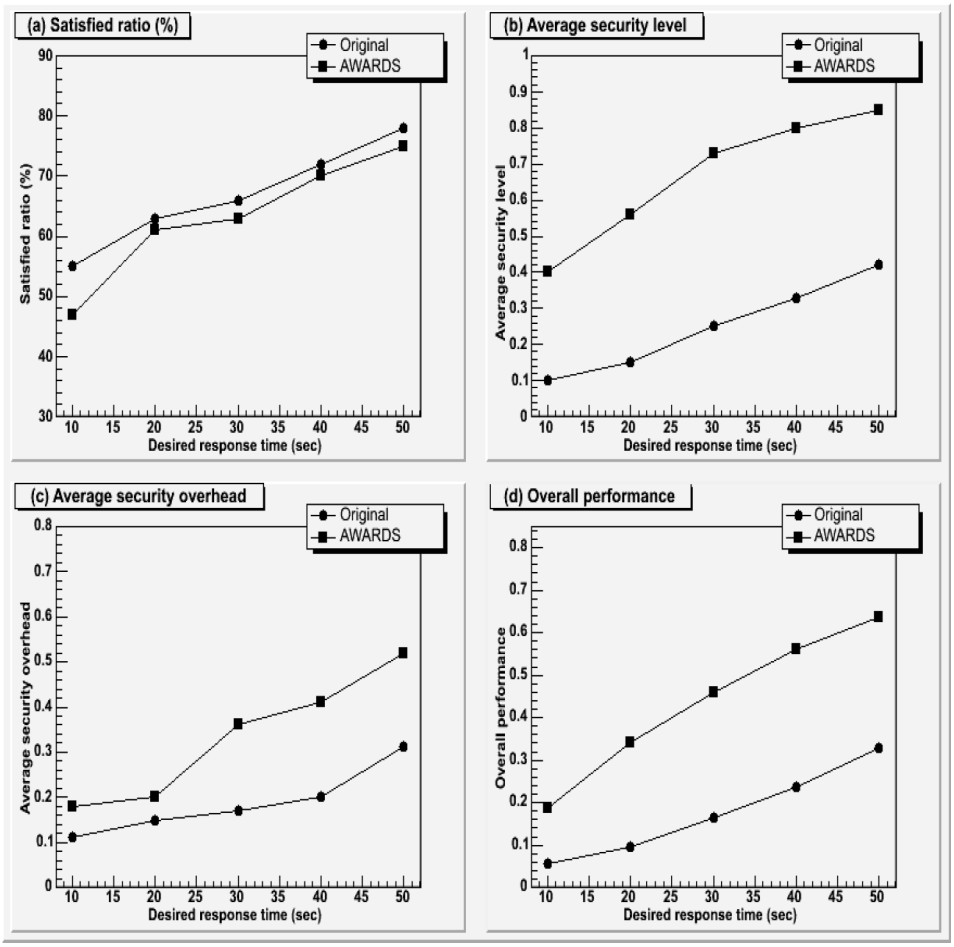

Fig. 10. Performance impact of desired response time. LU Decomposition.

with increasing write ratio because AWARDS aggressively enhances the security levels of write requests, which experience longer processing times due to higher security overheads (see Figure 9(c)). It is intriguing to observe from Figure 9(b) that the performance improvement of AWARDS in terms of security over the Original strategy become more pronounced for higher write ratios. The rationale behind this result is that AWARDS is conducive to the improvement in security for write requests and thus, an increasing number of write requests offers more opportunities for AWARDS to significantly improve security performance by choosing higher security levels for the write requests. Consequently, the overall performance improvement over the rival strategy is more striking for higher write ratios (see Figure 9(d)). This result suggests that the proposed AWARDS approach is suitable for improving the security of write-intensive applications like transaction processing, logfile updates, and data collection.

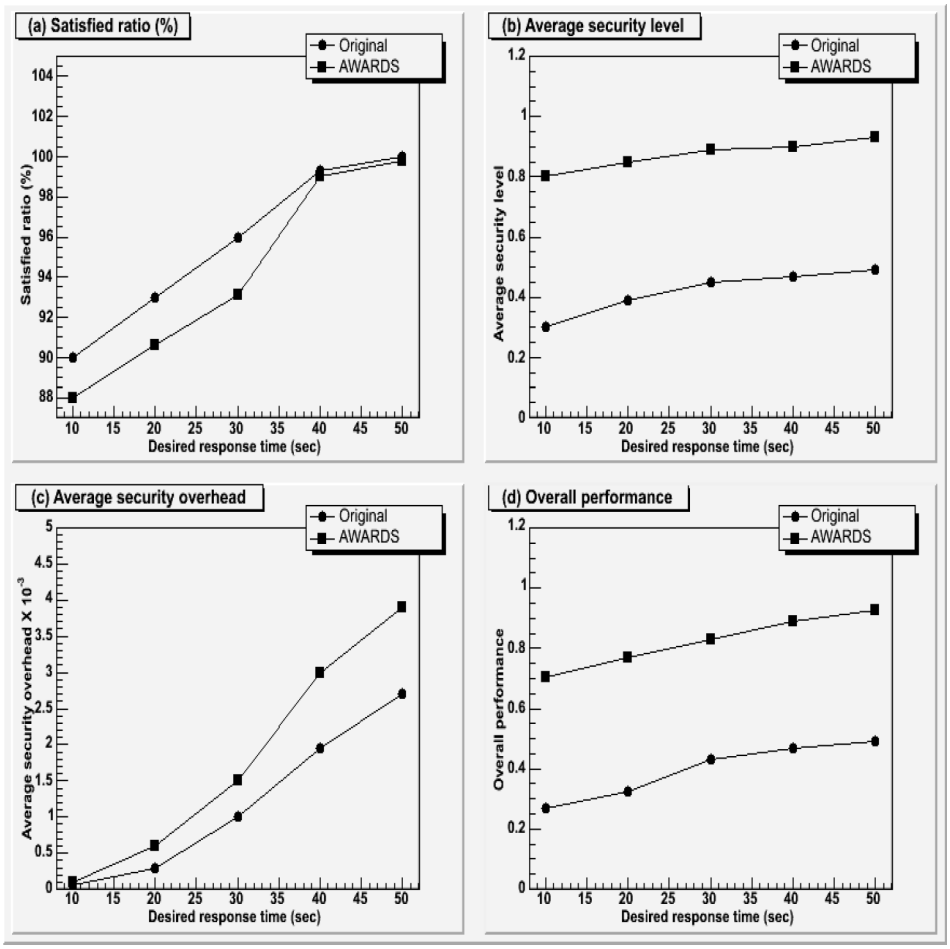

Fig. 11. Performance impact of desired response time. Sparse Cholesky.

7. REAL I/O-INTENSIVE APPLICATIONS

To validate the results from the synthetic workload, we used disk traces of real-world I/O-intensive applications to evaluate the performance of our strategy in comparison to the Original scheme. We chose two common I/O-intensive applications: LU decomposition [Hendrickson and Womble 1994] and sparse Cholesky [Acharya et al. 1996], which have different I/O patterns. The LU decomposition application tries to compute the dense LU decomposition of an out-of-core matrix, whereas the sparse Cholesky application is used to calculate Cholesky decomposition for sparse, symmetric positive-definite matrices.

First of all, we studied how desired response times affect satisfied ratios and security levels. Throughout this set of experiments, the disk bandwidth was set to 30MB/s. The results for the LU decomposition and sparse Cholesky applications are plotted in Figures 10 and 11, respectively. Figures 10(a) and 11(a) show that for the two I/O-intensive applications, satisfied ratios yielded

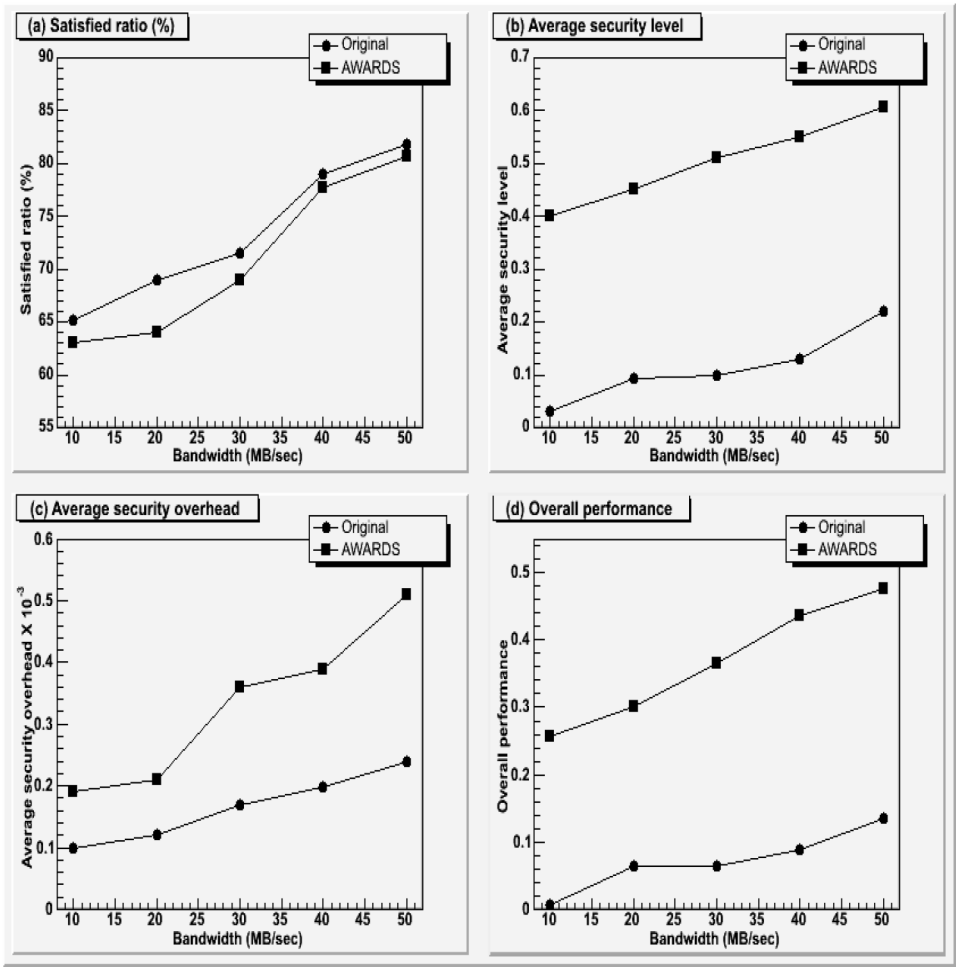

Fig. 12. Performance impact of disk bandwidth. LU decompostion.

by AWARDS are close to those of the Original strategy. This observation is especially true when the desired response time is long. Figures 10(b) and 11(b) demonstrate that as the desired response time increases, the average security levels for all cases increase. This is mainly because when the desired response time is enlarged, the possibility of improving security levels, without violating timing constraints, increases. This argument is supported by the results summarized in Figures 10(c) and 11(c), which reveal that the average security overheads for all cases grow with an increase in desired response times. Figures 10(d) and 11(d) summarize the overall performance of the two schemes. In general, the performance effect of the desired response depends in part on the applications. By comparing the two applications, we observe from Figures 10(d) and 11(d) that LU decomposition is more sensitive to desired response time, while sparse Cholesky is less sensitive. The cause of this performance difference can be explained as follows. Disk request arrival rate and

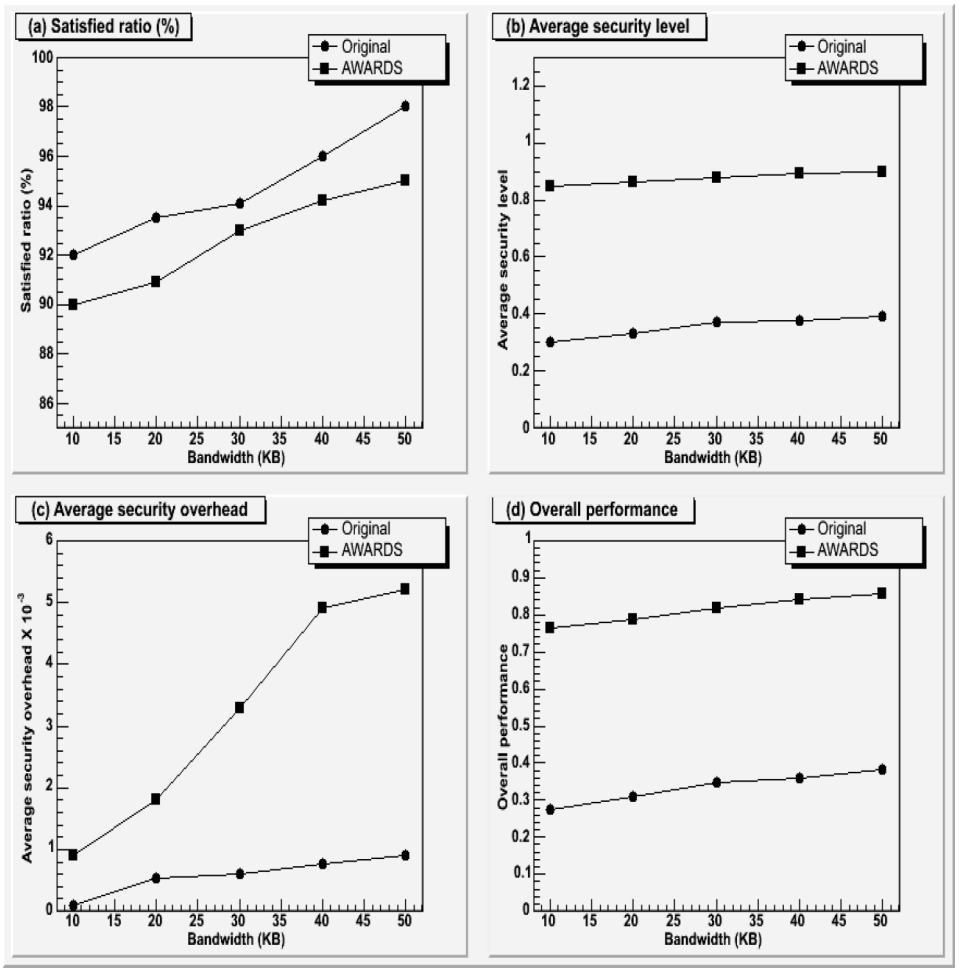

Fig. 13. Performance impact of disk bandwidth. Sparse Cholesky.

data size are two dominant factors for the satisfied ratio and average security level. A high arrival rate and large data size of disk requests give rise to LU decomposition's small satisfied ratios and average security levels (see Figures 10(a) and 10(b)). Consequently, the increase in desired response time provides more opportunities for LU decomposition to improve both the satisfied ratio and average security level, which in turn induce a more pronounced improvement in overall performance when the desired response time is enlarged.

Now we evaluate the impact of disk bandwidth on the two real applications. In this group of experiments, the disk bandwidth varies from 10 to 50MB/s with increments of 10MB/s. Figures 12 and 13 depict the disk bandwidth effect on AWARDS and Original strategies when LU decomposition and sparse Cholesky are used as the workload. Figures 12(a)–12(c) and 13(a–c) illustrate that for all the cases we have examined, increased disk bandwidth is a driving force of the improved satisfied ratios and average security levels. These results are

consistent with the disk bandwidth impact seen for the synthetic benchmarks in Section 6.3.

Not surprisingly, the disk bandwidth effect on the two strategies relies in part on the applications. More specifically, Figures 12(d) and 13(d) conclusively show that the overall performance improvement achieved by AWARDS is much more pronounced for the workload with the LU decomposition application as compared to the workload with sparse Cholesky. This can be partially attributed to the high sensitivity of LU decomposition to disk bandwidth. LU decomposition, compared with sparse Cholesky, is more sensitive to disk bandwidth because the relatively large arrival rate and data size of the workload with LU decomposition induce small satisfied ratios and average security levels, which allow more room for improvement in overall performance.

8. SUMMARY

In this article, we considered the flexible security requirements of disk write requests in the context of a local disk system. To protect stored data from being tampered with or disclosed, we proposed a security-aware storage system architecture. Next, we developed an adaptive write strategy for secure disk systems (AWARDS, for short). The AWARDS strategy can adaptively choose an appropriate security service for each write request in a way that maximizes the security of the local disk system, while making an effort to achieve the desired response times of all incoming disk requests. Further, we constructed an analytical model to estimate the probability that a disk request is completed before its desired response time. The model also can be used to derive the expected value of disk requests' security levels. We implemented a prototype of AWARDS and evaluated its performance using synthetic workloads, as well as two real-world I/O-intensive applications. Experimental results demonstratively show that our strategy outperforms an existing scheme in security and overall performance by up to 325.0% and 358.9% (with averages of 199.5% and 213.4%), respectively.

Currently, we are developing and evaluating an extended version of AWARDS for parallel disk systems. Both data placement and load balancing algorithms are being incorporated into the extended version.

ACKNOWLEDGMENTS

The work reported in this article was supported in part by the New Mexico Institute of Mining and Technology under Grant 103295 and by Intel Corporation under Grant 2005-04-070.

REFERENCES

ACHARYA, A., BENNETT, R., BEYNON, M., HOLLINGSWORTH, J., MENDELSON, A., SALTZ, J., SUSSMAN, A., AND UYSAL, M. 1996. Tuning the performance of I/O-Intensive parallel applications. In *Proceedings of the 4th Workshop on Input/Output in Parallel and Distributed Systems* (Philadephia, PA). 15–27.

Avitzour, D. 2004. Novel scene calibration procedure for video surveillance systems. *IEEE Trans. Aerospace and Electron. Syst. 40*, 3 (July), 1105–1110.

BALAFOUTIS, E., NERJES, G., MUTH, P., PATERAKIS, M., WEIKUM, G., AND TRIANTAFILLOU, P. 2003. Clustered scheduling algorithms for mixed-media disk workloads in a multimedia server. *J. Cluster Comput. 6,* 1, 75–86.

BLAZE, M. 1993. A cryptographic file system for UNIX. In *Proceedings of the ACM Conference on Communications and Computing Security*.

BORDAWEKAR, R., THAKUR, R., AND CHOUDHARY, A. 1994. Efficient compilation of out-of-core data parallel programs. Tech. Rep. SCCS-662, NPAC, Apr.

BOSCH, P. AND MULLENDER, S. J. 2000. Real-Time disk scheduling in a mixed-media file system. In *Proceedings of the Real-Time Technology and Applications Symposium*. 23–33.

BRUNO, J., GABBER, E., OZDEN, B., AND SILBERSCHATZ, A. 1999. Disk scheduling algorithms with quality of service guarantees. In *Proceedings of the IEEE Conference on Multimedia Computing Systems*.

CHANG, C., MOON, B., ACHARYA, A., SHOCK, C., SUSSMAN, A., AND SALTZ, J. 1997. Titan: A high-performance remote-sensing database. In *Proceedings of the 13th International Conference on Data Engineering*.

CHO, Y., WINSLETT, M., SUBRAMANIAM, M., CHEN, Y., KUO, S., AND SEAMONS, K. E. 1997. Exploiting local data in parallel array I/O on a practical network of workstations. In *Proceedings of the 5th Workshop on Input/Output in Parallel and Distributed Systems*. 1–13.

COFFMAN, J. R. AND HOFRI, M. 1990. Queueing models of secondary storage devices. In *Stochastic Analysis of Computer and Communication Systems,* H. Takagi, ed. North-Holland, Amsterdam.

DENNING, P. J. 1967. Effects of scheduling on file memory operations. In *Proceedings of the AFIPS Spring Joint Computer Conference*. 9–21.

DIMITRIJEVIC, Z. AND RANGASWAMI, R. 2003. Quality of service support for real-time storage systems. In *Proceedings of the International Conference on IPSI*.

FORNEY, B., ARPACI-DUSSEAU, A. C., AND ARPACI-DUSSEAU, R. H. 2001. Storage-Aware caching: Revisiting caching for heterogeneous storage systems. In *Proceedings of the International Symposium on File and Storage Technologies*.

HENDRICKSON, B. AND WOMBLE, D. 1994. The torus-wrap mapping for dense matrix calculations on massively parallel computers. *SIAM J. Sci. Comput. 15,* 5 (Sept).

HUBER, J., ELFORD, C. L., REED, D. A., CHIEN, A. A., AND BLUME, D. S. 1995. A high-performance portable parallel file system. In *Proceedings of the 9th ACM International Conference on Supercomputing*. 385–394.

HUGHES, J. AND CORCORNA, D. 1999. A universal access, smart-card-based, secure file system. *Atlanta Linux Showcase* (Oct.).

JACOBSON, D. AND WILKES, J. 1991. Disk scheduling algorithms based on rotational position. Tech. Rep. HPL-CSP-91-7, Feb.

LIGON, W. B. AND ROSS, R. B. 1996. Implementation and performance of a parallel file system for high-performance distributed applications. In *Proceedings of the IEEE International Symposium on High-Performance Distributed Computing*. 471–480.

MA, X., WINSLETT, M., LEE, J., AND YU, S. 2002. Faster collective output through active buffering. In *Proceedings of the International Symposium on Parallel and Distributed Processing*.

MAZIERES, D., KAMINSKY, M., KAASHOEK, M., AND WITCHEL, E. 1999. Separating key management from file system security. In *Proceedings of the ACM Symposium on Operating System Principles*.

MILLER, E., LONG, D., FREEMAN, W., AND REED, B. 2002. Strong security for distributed file systems. In *Proceedings of the Symposium on File Systems and Storage Technologies*.

NAHUM, E., O'MALLEY, S., ORMAN, H., AND SCHROEPPEL, R. 1995. Towards high-performance cryptographic software. In *Proceedings of the IEEE Workshop on the Architecture and Implementation of High-Performance Communication Subsystems*.

PRESLAN, K. W., BARRY, A. P., BRASSOW, J. E., ERICKSON, G. M., NYGAARD, E., SABOL, C. J., SOLTIS, S. R., TEIGLAND, D. C., AND O'KEEFE, M. T. 1999. 64-Bit, shared disk file system for Linux. In *Proceedings of the NASA Goddard Conference on Mass Storage Systems*.

QIN, X., JIANG, H., ZHU, Y., AND SWANSON, D. R. 2005. Improving the peformance of I/O-Intensive applications on clusters of workstations. *Cluster Comput. J. Netw. Softw. Tools Appl. 8,* 4 (Oct.).

REDDY, A. AND WYLLIE, J. 1999. Intergraded QoS management for disk I/O. In *Proceedings of the IEEE Conference on Mulitmedia Computing Systems*.

REIST, R. AND DANIEL, S. 1987. A continuum of disk scheduling algorithms. *ACM Trans. Comput. Syst.,* (Feb.), 77–92.

REUTHER, L. AND POHLACK, M. 2003. Rotational-Position-Aware real-time disk scheduling using a dynamic active subset. In *Proceedings of the IEEE Real-Time System Symposium*.

RIEDEL, E., KALLAHALLA, M., AND SWAMINATHAN, R. 2002. A framework for evaluating storage system security. In *Proceedings of the 1st Conference on File and Storage Technologies* (Monterey, CA).

SALEM, K. AND GARCIA-MOLINA, H. 1986. Disk striping. In *Proceedings of the 2nd International Conference on Data Engineering.* 336–342.

SCHEUERMANN, P., WEIKUM, G., AND ZABBACK, P. 1998. Data partitioning and load balancing in parallel disk systems. *VLD J*., (July), 48–66.

SELTZER, M., CHEN, P., AND OUSTERHOUT, J. 1990. Disk scheduling revisited. In *Proceedings of the USENIX Technical Conference.* 313 – 323.

SHENOY, P. AND VIN, H. 1998. Cello: A disk scheduling framework for next generation operating systems. In *Proceedings of the ACM SigMetrics Conference.*

SUMNER, T. AND MARLINO, M. 2004. Digital libraries and educational practice: A case for new models. In *Proceedings of the ACM/IEEE Conference on Digital Libraries*. 170–178.

TANAKA, T. 1993. Configurations of the solar wind flow and magnetic field around the planets with no magnetic field: Calculation by a new MHD. *J. Geophys. Res.,* (Oct.), 17251–17262.

XIE, T. AND QIN, X. 2005. A new allocation scheme for parallel applications with deadline and security constraints on clusters. In *Proceedings of the IEEE International Conference on Cluster Computing* (Boston).

XIE, T., QIN, X., AND SUNG, A. 2005. SAREC: A security-aware scheduling strategy for real-time applications on clusters. In *Proceedings of 34th International Conference on Parallel Processing* (Norway).

XIE, T. AND QIN, X. 2005. Enhancing security of real-time applications on grids through dynamic rescheduling. In *Proceedings of the 11th Workshop on Job Scheduling Strategies for Parallel Processing.*

YU, P. S., CHEN, M. S., AND KANDLUR, D. 1993. Grouped sweeping scheduling for DASD-Based multimedia storage management. *ACM Multimedia Syst. 1,* 3, 99–109.

Received June 2006; revised July 2006; accepted July 2006