Adaptive Quality of Security Control in Networked Parallel Disk Systems

Mais Nijim, Xiao Qin*, and Tao Xie
Department of Computer Science
New Mexico Institute of Mining and Technology
Soccoro, New Mexico 87801
{*mais, xqin, xietao*}@*cs.nmt.edu*

Abstract

Parallel disk systems, which have been widely used in building networked and data intensive applications, are highly scalable and can alleviate the problem of disk I/O bottleneck. Although a number of parallel disk systems have been developed, the systems lack a means to optimize quality of security for dynamically changing networked environments. We remedy this situation by proposing an adaptive quality of security control scheme for networked parallel disk systems (or ASPAD for short) that makes it possible for networked disk systems to adapt to changing security requirements and workload conditions. ASPAD is carried out in three phases: dynamic data partitioning, response time estimation, and adaptive security quality control. Hence, ASPAD is conducive to adaptively and expeditiously determining security schemes for disk requests in a way to improve security of networked parallel disk systems while making an effort to guarantee desired response times of the requests To prove the efficiency of the proposed approach, we simulate a networked parallel disk system into which nine cryptographic schemes are integrated. Empirical results show that ASPAD significantly improves overall performance over an existing strategy with an average of 65%.

1 Introduction

In the past ten years, disk systems have been widely used in building networked and data intensive applications, including but not limited to video surveillance [1], and remote-sensing database systems [2]. Parallel disk systems are important to these applications, because parallel disk systems are highly scalable and can alleviate the problem of disk I/O bottleneck. To exploit I/O parallelism in parallel disk systems, we have to partition and distribute data among an array of disks. Disk I/O parallelisms can be provided in forms of inter-request and intra-request parallelism. Inter-request parallelism allows multiple independent requests to be served simultaneously by a parallel disk system, whereas intra-request parallelism enables a single disk request to be processed by multiple disks in parallel. A parallelism degree of a data request is the number of disks where the requested data resides [11].

Many data-intensive applications embrace a rich variety of security services to protect data residing in disk systems from talented intruders. Further, it is often desirable for next generation parallel disk systems to be highly flexible in order to support varying quality of security at different times during a data intensive application lifetime. This trend is especially true for data-intensive applications where disk requests need to be completed within specified response times [3]. Thus, high quality of security and guaranteed response times are two major performance goals to be achieved by parallel disk systems. As such, the necessity of automatic tuning cryptographic schemes and parallelism degrees are increasingly becoming critical and challenging issues in development of modern parallel disk systems.

In this paper, we propose an adaptive quality of security control mechanism for parallel disk systems. The security control mechanism aims at making it possible for parallel disk systems to adapt to changing security requirements and workload conditions, thereby providing a rich variety of data storage environments for data-intensive applications. To achieve this design goal, we developed an adaptive strategy for parallel disk systems (or ASPAD for short) at the heart of the control mechanism. The ASPAD strategy endeavors to determine cryptographic schemes for disk requests in a way to improve security of parallel disk systems while making the best effort to guarantee desired response times of the requests.

The rest of the paper is organized as follows. In Section 2, we summarize related work. Section 3 details a model of disk requests and the new architec-

*Contact author. http://www.cs.nmt.edu/∼xqin

ture of storage systems. In section 4, we propose the adaptive write strategy for security-aware storage systems. Section 5 present the experimental results based on both synthetic benchmarks (read/write) and real data-intensive applications. Finally, Section 6 concludes the paper with future directions.

2 Related Work

In the last decade, much attention has been paid to the issue of security in storage systems. Riedel *et al.* proposed a framework of core functions that required for secure storage systems [9]. Protecting data in untrusted storage systems is of critical importance and, therefore, a number of cryptographic file systems have been implemented in a way that data is stored in encrypted form [4]. Although an array of secure services have been implemented for storage systems, there is a lack of adaptive flexibility to choose appropriate cryptographic schemes to meet disk requests' dynamic security requirements.

Disk I/O has become a performance bottleneck for data-intensive applications due to the widening gap between processor speeds and disk access speeds [8]. To help alleviate the problem of disk I/O bottleneck, a large body of work has been done on parallel disk systems. Kallahalla and Varman designed an on-line buffer management and scheduling algorithm to improve performance of parallel disks [5]. Kotz and Ellis proposed investigated several write back policies used in a parallel file system implementation [6]. Our approach is fundamentally different from the aforementioned techniques in that we focused on optimizing quality of security for parallel disk systems. Further, our strategy is orthogonal to the existing techniques in the sense that our scheme can be readily integrated into existing parallel disk systems to substantially improve security of the systems.

In our previous work, we proposed an array of security-aware scheduling algorithms for clusters [11]; moreover, we recently developed an adaptive write strategy for local disk systems [7]. Since these approaches limit their applicability to either computing resources or a single disk system, our previous schemes are unable to be employed to parallel disk systems.

3 Architecture and Disk Requests

3.1 Architecture

In our study we consider a security-aware networked parallel disk system, which encompasses a parallel disk system, network, security service middleware, a data partitioning mechanism, an adaptive security service controller. The architecture of the networked parallel disk system is delineated in Fig.1.

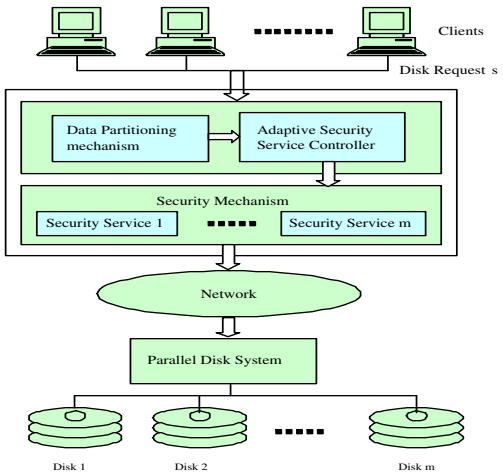

Figure 1: **Architecture of a security-aware networked parallel disk system**

Note that this architecture is general enough to accommodate a broad range of storage systems, including, of course, network attached storage devices and storage area networks.

The data partitioning mechanism, security mechanism, and security service controller are at the heart of the proposed architecture. The data partitioning mechanism is geared to divide a large amount of data into fixed-size of data units stored on a number of disks. We consider file striping in this study, because it is a generic method for a wide variety of data type [11]. The security mechanism features an array of cryptographic schemes to support read and write requests issued by clients. The security service controller is developed to make a decision of which security service is chosen to protect data for each request, thereby achieving the best tradeoff between security and performance.

In the proposed architecture, the clients issue read and write requests to the parallel disk system through network links. With respect to data partitioning and security service control, read and write requests are treated in different ways. Data partitioning and security service selections for read requests are carried out statically, since static partitioning results in balanced load across all the disks in the system. In contrast, dynamic data partitioning is performed for write requewwwdu sts to alleviate imbalanced load among the disks due to skewed access frequencies. The security service controller selects the most appropriate security service to protect data for each write request. In this paper, we restrict our attention to adaptive quality of security control for write requests. Through-

out this paper, disk requests issued by clients are write unless otherwise specified. After the process of data partitioning and security service controlling is accomplished for a write, the striping units of data in encrypted form are transferred through the network links. Finally, the striping units are written to multiple disks in parallel.

3.2 Quality of security

Recall that the security-aware networked parallel disk system encompasses an array of security service providing various quality of security. Each security service can be characterized by its security quality measured by security level ranging from 0.1 to 1.0 [7]. Thus, the higher the security level of a service, the better the security quality of the service. cryptographic schemes generally fall into three categories: confidentiality, integrity, and availability. Without loss of generality, in our quality of security model we address the confidentiality issues by employing nine cryptographic algorithms in our framework [7]. Note that the quality of security model can be easily extended to incorporate the other two security service categories. We assume that the clients, network, and parallel disk system are always available by the virtue of fault-tolerant mechanisms residing in these components. This assumption is valid, because the overhead of supporting reliability in the system can be envisioned as a part of security overhead.

Security overheads incurred by the cryptographic algorithms depend on size of data to be encrypted and performance of the algorithms, each of which is assigned a security level. Let σ_i denote the security level of a cryptographic algorithm used to encrypt the data for disk request r_i, and d_i is the size of data to be encrypted. We can obtain the security overhead of request r_i using Eq. 1, where $P(\sigma_i)$ is a function mapping security level i to the performance (measured by KB/ms) of the corresponding confidentiality service [7].

$$T_{security}(\sigma_i, d_i) = \frac{d_i}{P(\sigma_i)} \quad (1)$$

3.3 Disk requests with security and performance requirements

We consider in this study data-intensive applications with both security and performance constraints, meaning that disk requests issued by the applications to a parallel disk system impose both security and performance requirements. While the security requirement of a request, e.g., r_i, is specified by a low bound s_i on security level that the networked disk system has to provide. Hence, the security service controller must ensure that is greater than or equal to s_i. In this paper, we investigate desired response time, which is a specific performance requirement; and the desired response time of request r_i is represented by t_i. We denote parallelism degree of r_i by p_i. It is worth noting that parallelism degrees play a critical role in performance tuning of networked parallel disk systems. As such, it is appealing to devise the data partitioning mechanism to automatically determine a parallelism degree for each request in a way to improve throughput of the system. Given parallelism degrees of requests, quality of security for the requests can be tuned in a judicious manner by the security service controller.

The first step toward improving quality of security is to quantitatively measure security benefits of a disk request. To achieve this goal, we calculate the security benefit of request r_i using the following security level function.

$$S(r_i) = \sum_{j=1}^{p_i} \sigma_{ij}, \sigma_{ij} \geq s_i \; and \; p_i \leq m \quad (2)$$

where m is the number of disks in the parallel disk system and σ_{ij} is the security level of a confidentiality service chosen for the jth striping unit of r_i.

Given a sequence of requests $R = \{r_1, r_2, ..., r_n\}$ we can obtain the security benefits experienced by the requests. Thus, we have

$$S(R) = \sum_{i=1}^{n} S(r_i) \quad (3)$$

Now we obtain the following non-linear optimization problem formulation to compute the optimal security benefit of the networked parallel disk system

$$maximize \;\; S(R) = \sum_{i=1}^{n} \sum_{j=1}^{p_i} \sigma_{ij}$$

subject to

$$(a) \;\; \forall 1 \leq i \leq n : max_{1 \leq j \leq p_i}\{\theta_{ij}\} \leq t_i,$$

$$(b) \;\; \sigma_{ij} \geq s_i \; and \; p_i \leq m \quad (4)$$

where θ_{ij} is the response time for the jth striping unit of request r_i. In an effort to enhance security of the system, we have to guarantee that the following three conditions are met. First, the response time of all striping unit in request r_i must be smaller than the desired response time. Second, the low bound on security level can not be violated. Third, the parallelism degree of r_i has to be smaller than or equal to the number of disks in the system.

```
Input: r: a newly arrived disk request
       t_i: desired response time of the ith request
       s_i: the ith request's lower bound on security level
       Q, a waiting queue at the client side
1. Insert r into Q based on the earliest desired response time first policy
2. for each request r_j in the waiting queue Q do
     /* Phase 1: dynamic data partitioning */
3.   Calculate the optimal parallelism degree p_i of r_i
4.   Partition the request into p_i stripe units
5.   for each stripe unit of r_i do
6.     Initialize the security level σ_ij of the jth stripe unit to the minimal value s_i
       /* Phase 2: response time estimation */
7.     Estimate the response time of the jth stripe unit
       /* Phase 3: adaptive security quality control */
8.     while estimated response time < desired response time t_i do
9.       if σ_ij < 0.9 then /* σ_ij can be further increased */
10.        Increase security level σ_ij by 0.1
11.        Estimate the response time of the jth stripe unit
12.      else break  /* σ_ij can not be further increased */
13.    end while
14.    if estimated response time > desired response time t_i then
15.      Decrease security level σ_ij by 0.1;
16.    Apply the security service with level σ_ij to the jth stripe unit
17.    Deliver the jth unit through the network subsystem to the disk subsystem
18. end for
```

Figure 2: **The ASPAD algorithm for networked parallel disk systems**

4 The ASPAD Algorithm

We developed an adaptive quality of security control algorithm for networked parallel disk systems. The ASPAD algorithm is geared to adaptively determine the most appropriate encryption algorithm for stripe units of a disk request while guaranteeing the desired response time of the request. Specifically, the algorithm is carried out in three phases: dynamic data partitioning, response time estimation, and adaptive security quality control. To heuristically improve security of the networked disk systems, ASPAD endeavours to minimize the response time of a request. Hence, the first phase dynamically calculates the optimal parallelism degree of the request, thereby reducing delays at the parallel disk subsystems. During the second phase of the algorithm, the response time of each stripe unit is estimated. Phase three, guided by the estimated response time obtained and desired response time, optimizes the security level of each stripe unit provided that the request's response time does not exceed the request's desired response time. The complete algorithm for quality of security control is outlined in Fig. 2. When a request is issued by a client to the system, ASPAD inserts the newly arrived requests into the waiting queue based on the earliest desired response time first policy (see Step 1). After the data partitioning of each request in the queue, ASPAD initializes the security levels of all stripe units of request r_i to the minimal levels s_i (see Step 6). In an effort to gradually increase the security levels of stripe units, ASPAD guarantees that all requests will be completed before their desired response times. Thus, the following property needs to be satisfied in ASPAD.

Property 1. With respect to the ith request, the following two conditions must hold if the jth stripe unit's security level is increased by 0.1:
(1) The current security level σ_{ij} is less than 0.1;
(2) $T_j(r_i, p_i, \sigma_i) \leq t_i$, where T_j is the response time of the jth stripe unit, t_i is the desired response time of the request, and $T_j(r_i, p_i, \sigma_i) = T_{queue} + T_{partition} + T^{ij}_{proc}(r_i, p_i, \sigma_{ij})$.

Steps 10-11 are repeatedly performed to optimize security levels until a request's desired response time can not be guaranteed (see Step 12) or the security levels are approaching 1.0. Consequently, the ASPAD algorithm dramatically increases the security levels while making the best effort to finish all the disk requests before their desired response times. The time complexity of ASPAD is evaluated (see Theorem 1) as the number of waiting requests and maximum parallelism degree vary.

Theorem 1. The time complexity of ASPAD is *O(np)*, where n is the number of disk requests in the waiting queue, p is the maximum parallelism degree.

Proof. It takes *O(10)* time to increase the security level of each stripe unit (see Step 8). Since there are *O(p)* number of stripe units in each disk request, the time complexity of optimizing security levels of a write request is *O(10p) = O(p)*. There are n disk requests in the waiting queue and, therefore, the time complexity of improving security of all the requests is: *O(n)O(p) = O(np)*.

Compared with processing times at the network subsystems and parallel disk subsystem, the overhead of ASPAD can be negligible. This argument is especially true for workload conditions where arrival rates of disk requests are relatively low, because the number of waiting disk requests in these cases is small.

5 Evaluation

To evaluate the performance of the ASPAD strategy in an efficient way, we simulated a parallel disk system, into which nine encryption services were integrated. Table 2 summarizes important system parameters used to resemble real world disks. In addition, we implemented a data-partitioning algorithm to optimize parallelism degrees of large disk I/O requests. We will first compare the performance of a parallel disk system with ASPAD with that of another system without employing adaptive quality of security control mechanism. We will then study effects of varying arrival rates on the performance of the two disk systems. Next, we will compare and evaluate the two disk sys-

Table 1: Characteristics of the simulates disk system

Number of disks	4
Block size	1 KB
Number of tracks per cylinder	11
Number of Cylinder per disk	1435
Capacity of one disk	539 MB
Average seek time	12ms
Blocks revolution per minute	4400 RPM
Transfer rate per disk	2.44 MB/s

tems with respect to security requirements imposed by disk requests. Finally, we will also analyze performance impacts of parallelism degrees on the parallel disk systems.

In our simulation experiments, we made use of the following three metrics to demonstrate the effectiveness of the ASPAD scheme. (1) *Satisfied ratio* is a fraction of total arrived disk requests that are found to be finished before their desired response times. (2) *Security level* is a sum of security level values of all disk requests issued to the parallel disk systems. (3) *Overall performance* is performance metric measured by a product of the satisfied ratio and the security level.

5.1 Impacts of arrival rate

This experiment is aimed at comparing the ASPAD strategy with a baseline scheme that makes no use of ASPAD. With different settings of parallelism degrees and data sizes, we study the impacts of varying arrival rates on system performance. To achieve this goal, we increased the arrival rate of I/O requests from 0.1 to 0.5 No./Sec. while setting the parallelism degree to 3 and data size to 100KB, 10MB, and 100 MB, respectively. Fig. 3 plots the satisfied ratios, security levels, and overall performance of the parallel disk systems with and without ASPAD. Figs 3(aa), (ba) and (ca) reveal that the ASPAD strategy yields satisfied ratios that are very close to those of the parallel disk system making no use of ASPAD. This is mainly because ASPAD endeavors to guarantee timing constraints of disk requests while maximizing security of the parallel disk system. Figs 3(ab), 3(bb), and 3(cb) illustrate that ASPAD significantly improves the quality of security of the disk system by an average of 64%. We can attribute the improvement in security to the fact that ASPAD strives to increase security level of each parallel disk request provided that the corresponding real-time requirement can be met. It is observed that as the value of arrival rate increases, the security levels

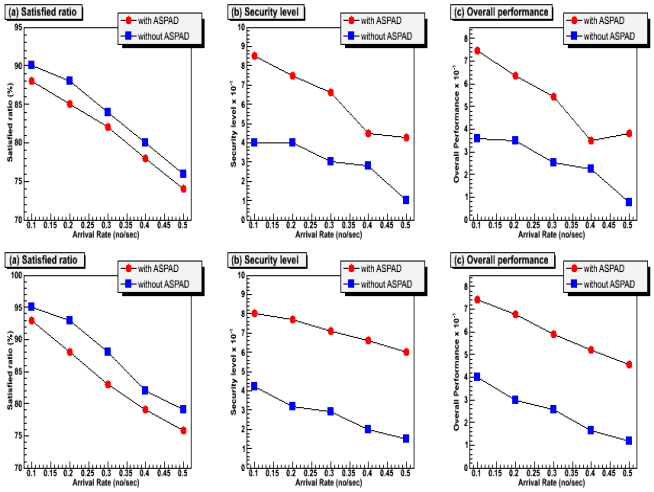

Figure 3: **Impact of Arrival rate**

of the both systems decrease. This result is not surprising because high arrival rates lead to high workload, forcing the parallel disk systems to merely meet the minimal security requirements of a vast majority of requests in order to process a large number of requests in a timely manner. Interestingly, ASPAD always achieves higher security levels compared with the parallel disks without ASPAD. It is worth noting that the security improvement comes at the cost of satisfied ratios (see Figs 3(aa), 3(ba) and 3(ca), because the satisfied ratios of ASPAD are slightly reduced due to relatively high security overhead caused by the ASPAD strategy. Figs 3(ac), 3(bc), and 3(cc) reveal that ASPAD substantially boosts the overall performance. The reason of the expected overall performance improvement is two-fold. First, ASPAD adaptively enhances the security levels for disk I/O requests. Second, the performance gains in security level can eventually offset the extra security overhead.

5.2 Impacts of parallelism degree

Disk I/O parallelisms are implemented in forms of inter-request and intra-request parallelism. Without loss of generality, in this study we considered the intra-request parallelism. Nevertheless, the proposed ASPAD strategy can be readily applied to disk workload conditions with inter-request parallelism.

Now we focus on the effects of parallelism degree on the performance of ASPAD and the alternative. In this set of experiments, the parallelism degree was varied from 2 to 16. Fig. 5a shows that when the parallelism degree increases, the performance in terms of satisfied ratio is improved. The rationale behind this

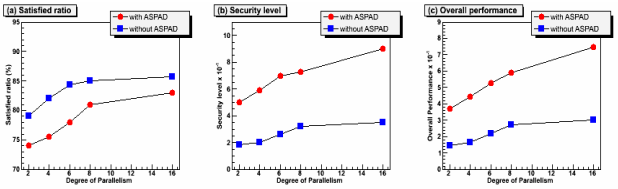

Figure 4: **The impact of the degree of parallelism when arrival rate = 0.5 No./sec.**

observation is that increasing parallelism degrees indicates the increasing number of disks over which a request is served (thereinafter referred to as striping width) and, thus, increasing the striping width has a strong likelihood to reduce response times of the requests and enhance throughput of the parallel disk systems. It is intriguing to observe from Fig. 5b that the high parallelism degrees give rise to high levels of security. We attribute this performance trend to positive impacts of the large striping widths, which substantially decrease the response times. Thus, the decrease in the response times ultimately makes it possible for an increasing number of disk requests to be completed before their desired response times. As shown in Fig. 5c, the overall performance improvement of ASPAD over the alternative is more prominent when the parallelism degree becomes high. This is because, first, ASPAD's performance degradation in satisfied ratio becomes less significant as the parallelism degree increases (see Fig. 5a). Second, the security level discrepancy between ASPAD and the competitive scheme is widened as the parallelism degree rises, meaning that high parallelism degrees offer more opportunities for ASPAD to increase security levels of disk requests in a judicious manner.

6 Summary

High quality of security and guaranteed response times are two major performance goals to be achieved by parallel disk systems. The reason is two-fold. First, it is often desirable for next generation parallel disk systems to be highly flexible in order to support varying quality of security at different times during a data intensive application lifetime. Second, this trend is especially true for data-intensive applications where disk requests need to be completed within specified response times. In this paper, we have proposed and evaluated an adaptive quality of security control scheme for networked parallel disk systems (or ASPAD for short) to protect information in data-intensive applications from being tampered by talented intruders. ASPAD aims at adapting to changing security requirements and workload conditions in the context of networked parallel disk systems. To achieve this goal, ASPAD is carried out in three phases: dynamic data partitioning, response time estimation, and adaptive security quality control. Specifically, ASPAD determines the most appropriate cryptographic schemes for disk requests issued by clients, thereby improving security for networked parallel disk systems and guaranteeing desired response times for the requests. We simulated a networked parallel disk system where a data partitioning method was implemented to find optimal values of parallelism degrees. Experimental results demonstratively show that ASPAD significantly outperforms an existing scheme that does not employ the adaptive quality of security controller.

Several important problems remain open. For example, we have ignored network delays in the data partioning method. Our future work will focus on impacts of network delays on performance of ASPAD. Further, we will integrate ASPAD with fault tolerance techniques to provide high availability for critical data-intensive applications.

References

[1] D. Avitzour, "Novel scene calibration procedure for video surveillance systems," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 40, No. 3, pp. 1105-1110, July 2004.

[2] C. Chang, B. Moon, A. Acharya, C. Shock, A. Sussman, and J. Saltz. "Titan: a High-Performance Remote-Sensing Database, *Proc. the 13th Int'l Conf. Data Engineering*, Apr 1997.

[3] Z. Dimitrijevic and R. Rangaswami, "Quality of Service Support for Real-time Storage Systems," *Proc. Int'l Conf. IPSI*, Sv. Stefan, Montenegro, October 2003.

[4] J. Hughes and D. Corcorna, "A Universal Access, Smart-Card-Based, Secure File System," *Atlanta Linux Showcase*, Oct. 1999.

[5] M. Kallahalla and P. J. Varman, "Improving parallel-disk buffer management using randomized writeback," *Proc. Int'l Conf. Parallel Processing*, pp. 270-277, Aug. 1998.

[6] D. Kotz and C. Ellis, "Cashing and writeback policies in parallel file systems," *Proc. IEEE Symp. Parallel and Distributed Processing*, pp. 60-67, Dec. 1991.

[7] M. Nijim, X. Qin, T. Xie, and M. Alghamdi, "Awards: An Adaptive Write Scheme for Secure Local Disk Systems," *Proc. 25th IEEE Int'l Performance Computing and Communications Conference*, Phoenix, AZ, April 2006.

[8] S. Rajasekaran, "Selection algorithms for parallel disk systems," *Proc. Int'l Conf. High Performance Computing*, pp.343-350, Dec. 1998.

[9] E. Riedel, M. Kallahalla, and R. Swaminathan, "A Framework for Evaluating Storage System Security," *Proc. the 1st Conf. File and Storage Technologies*, Monterey, CA, Jan. 2002.

[10] T. Xie, X. Qin, and M. Nijim, "SHARP: A New Real-Time Scheduling Algorithm to Improve Security of Parallel Applications on Heterogeneous Clusters," *Proc. 25th IEEE Int'l Performance Computing and Communications Conf.*, Phoenix, AZ, April 2006.

[11] P. Scheuermann, G. Weikum, and P. Zabback, "Data partioning and load balancing in parallel disk systems," *VLDB Journal*, Vol.7, pp.48-66, 1998.