Power: A Metric for Evaluating Watermarking Algorithms (CERIAS TR 2001-55, extended abstract) *

Radu Sion
Computer Sciences, Purdue University,
Center for Education and Research
in Information Assurance and Security (CERIAS)
West Lafayette, IN, 47907, USA
http://www.cs.purdue.edu/homes/sion
sion@cs.purdue.edu

Mikhail Atallah
CERIAS,Computer Sciences Purdue University
http://www.cs.purdue.edu/homes/mja
mja@cs.purdue.edu

Sunil Prabhakar
CERIAS,Computer Sciences Purdue University
http://www.cs.purdue.edu/homes/sunil
sunil@cs.purdue.edu

Abstract

An important parameter in evaluating data hiding methods is hiding capacity *[3] [11] [12], [16] i.e. the amount of data that a certain algorithm can "hide" until reaching allowable distortion limits.*

One fundamental difference between watermarking [4] [5] [7] [8] [9] [13] [14] [17] [18] [19] [20] [21] [22] [23] and generic data hiding resides exactly in the main applicability and descriptions of the two domains. Data hiding aims at enabling Alice and Bob to exchange messages [1] [6] [15] in a manner as resilient and stealthy as possible, through a medium controlled by evil Mallory. On the other hand, digital watermarking is deployed by Alice to prove ownership over a piece of data, to Jared the Judge, usually in the case when Tim the Thief [1] *benefits from using/selling that very same piece of data (or maliciously modified versions of it).*

In the digital framework, watermarking algorithms that make use of information hiding techniques have been developed and hiding capacity was naturally used as a metric in evaluating their power to hide information.

Whereas the maximal amount of information that a certain algorithm can "hide" (while keeping the data within allowable distortion bounds) is certainly

*Portions of this work are supported by Grants IIS-9985019, EIA-9903545 from the National Science Foundation, and by sponsors of the Center for Education and Research in Information Assurance and Security. This work has appeared in the Proceedings of the IEEE International Conference on Information Technology: Coding and Computing, Las Vegas, Nevada, April 2002, Computer Society Press

[1] Tim's middle name is *Mallory*.

related to the ability to assert ownership in court, it is not directly measuring its "power of persuasion" (i.e. how convinced will Jared the Judge and the Jury be when presenting the algorithm and applying it on an object in dispute), in part also because it doesn't consider directly the existence and power of watermarking attacks.

*In this paper we show why, due to its particularities, watermarking requires a different metric, more closely related to its ultimate purpose, claiming ownership in a court of law. We define one suitable metric (*watermarking power*) and show how it relates to derivates of hiding capacity. We prove that there are cases where considering hiding capacity is sub-optimal as a metric in evaluating watermarking methods whereas the metric of* watermarking power *delivers good results.*

1 Introduction

Defining a unified watermarking evaluation metric is a non-trivial task. Most domain specific metrics are derived from the concept of watermarking capacity and do not directly relate to the main purpose of watermarking per se, i.e. claiming ownership in court. Theoretical approaches [3] [10] [11] [12] [16] explore the broader area of steganography and information hiding in a generic manner.

Proof of ownership is usually achievable by demonstrating that the particular piece of data exhibits a certain rare property (read "hidden message" or "watermark"), usually known only to Alice (with the aid of a "secret" - read "watermarking key"), the property being so rare that if one considers any other random piece of data similar (in terms of usability, see below) to the one in question, this property is "very improbable" to apply.

There is a threshold determining Jared's convinceability related to the "very improbable" assessment. Nevertheless this defines a main difference from steganography: Jared doesn't care what the property is, as long as Alice can prove it is she who embedded/induced it to the original (non-watermarked) data object.

It is to be stressed here that another particularity of watermarking is the emphasis on 'detection' rather than 'extraction'. Extraction of a watermark (or bits of it) is usually a part of the detection process but just complements the process up to the extent of increasing the ability to convince in court. If recovering the watermark data in itself becomes more important than detecting the actual existence of it (aka. 'yes/no answer') then this is a drift towards covert communication and pure steganography.

The paper is structured as follows. Section 2 formalizes some of the main concepts such as *watermark*, *usability*, *usability domains*, *watermarking algorithm* and defines our metric, *watermark power*. Section 3 presents a simple scenario where metrics derived from hiding capacity perform poorly in evaluating a certain watermarking method whereas the concept of *power* delivers good results. Section 4 defines future envisioned research issues.

2 Model and Definitions.

Let $\mathbb{D}$ be the domain of all possible data objects to be considered for watermarking.

Considering any reasonable security assumptions and attacks, it becomes clear that a correct watermarking algorithm has to assure that the domain of all possible watermarked data objects (i.e. results from watermarking objects in $\mathbb{D}$) should be a subset of $\mathbb{D}$. For simplicity we assume that any considered algorithm produces watermarked objects only in $\mathbb{D}$ or that $\mathbb{D}$ is simply the union of all the closures over $\mathbb{D}$ of all resulting watermarked objects from considered algorithms.

For example in case of digital media objects we can simply assume that $\mathbb{D}$ is the set of all variable sized bit strings over $\mathbb{B} = \{0, 1\}$.

Objects $d \in \mathbb{D}$ have associated values induced by the object creator. Watermarking tries to protect this association between the value carrying object and its creator. Complex objects can exhibit different value levels when put to different uses. We need a way to express the different associated values of objects, in different *usability domains*.

Usability Domain: A *usability domain* is defined as a set of functionals, $e = \{f | f : \mathbb{D} \to [0, 1]\}$, quantifying value in terms of usability. In a real world algorithm the considered usability domain is constructed by mapping real world properties to actual parametrized functionals in e. Also most likely $|e| = 1$, that is, each domain contains only one significant function of usability. Notation: Let the set of all usability domains be $\mathbb{U}$.

Usability: The *usability* of an object $d \in \mathbb{D}$ corresponding to a certain domain $e \in \mathbb{U}$ ($e = \{f_1, f_2, ..., f_q\}$) is defined as $u(d, e)$ where $u : \mathbb{D} \times \mathbb{U} \to [0, 1]$. $u(d, e)$ is a combination of all the elements of e. For simplicity we will assume that $|e| = 1$. In this case we define $e = \{u\}$.

The concept of usability enables the definition of a certain threshold below which the object is not "usable" anymore in the given domain. In other words, it "lost its value" to an unacceptable degree. The notion of usability is related to *distortion*. A highly distorted object (e.g. as result of watermark embedding or attacks) will likely suffer a drop in its distortion domain usability.

For simplicity, in the following we consider a single usability domain $e \in \mathbb{U}$, unless otherwise specified.

Change in Usability: The *difference in usability* is defined as $\Delta u : \mathbb{D} \times \mathbb{D} \times \mathbb{U} \to [-1, 1]$, where $\Delta u(d_1, d_2, e) = u(d_1, e) - u(d_2, e)$. This quantity is easier to derive from a real world mapping and has a higher impact on the actual embedding decisions made.

Usability Vicinity: Let $V \subset \mathbb{U}$ be a set of usability domains and a maximum allowed difference in usability Δu_{max}. Then we say that element $x \in \mathbb{D}$ is *in the radius* Δu_{max} *usability vicinity of* $d \in \mathbb{D}$ *with respect to* V [2] if and only if $\forall v \in V$, $\Delta u(d, x, v) < \Delta u_{max}$.

Note that the usability vicinity of a certain object $d \in \mathbb{D}$ with respect to a considered set of usability domains $V \subset \mathbb{U}$ defines actually the set of possible watermarked versions of d with respect to V and Δu_{max}.

Watermark: Given an un-marked object $d \in \mathbb{D}$ and the considered watermarked version of it, $d' \in \mathbb{D}$, where d' is within radius Δu_{max} usability vicinity of d, a key $k \in \mathbb{K}$ [3], a set of usability domains $V \subset \mathbb{U}$ and a maximum allowed difference in usability Δu_{max}, a *watermark* can be asserted by a special property functional $w : \mathbb{D} \times \mathbb{K} \to \mathbb{B}$, defined by the following:

[2] And vice-versa. It is commutative.

[3] Where $\mathbb{K}$ is the set of secrets (i.e. keys) involved in the watermarking process.In many cases we assume that $\mathbb{K}$ is also a subset of all variable sized binary bit strings or can be easily mapped to one. Sometimes $\mathbb{K} \subset \mathbb{D}$.

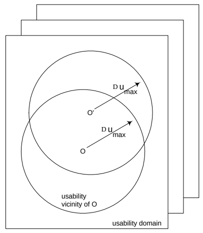

Figure 1. Usability vicinities for the original object O and for the watermarked object O' overlap. A good watermarking algorithm minimizes the size of the overlap.

$w(d,k) = 0$, $w(d',k) = 1$ and there exists $\epsilon \in (0,1)$ such that the probability $P(w(x,k) = 1 | w(d',k) = 1) < \epsilon$, for any $x \in \mathbb{D}, x \neq d'$, inside the radius Δu_{max} usability vicinity of d (1). [4] Notation: Let $\mathbb{W}_{\mathbb{D}}$ be the set of all w over a given $\mathbb{D}$.

In plain words, a watermark can be defined as a special induced (through watermarking) property (w) of a certain watermarked object $d' \in \mathbb{D}$, so rare, that if we consider any other object $x \in \mathbb{D}$, with a "close-enough" usability level with the original object d, the probability that x exhibits the same property can be upper-bounded.

Note: Intuitively, one main challenge of watermarking is to find/derive d' such that, given only d' it will be very hard for an attacker to determine an $x \neq d'$ inside the usability vicinity of d.

Algorithm: A *watermarking algorithm* can be described as a functional $a : \mathbb{D} \times \mathbb{K} \rightarrow \mathbb{D} \times \mathbb{W}$, which, given as input an object $d \in \mathbb{D}$ provides a watermarked version of the object, d', and an associated property functional w that enables watermark detection. Notation: Let $\mathbb{A}_{\mathbb{D}}$ be the set of all a over a given $\mathbb{D}$.

Attack: Given a watermarking algorithm $a \in \mathbb{A}_{\mathbb{D}}$, an object $d \in \mathbb{D}$ and its watermarked version $d' \in \mathbb{D}$, Δu_{max}, ϵ_{max}, $\forall k \in \mathbb{K}$ and noting $a(d,k) = (d',w)$, a *watermarking attack* is defined by $z : \mathbb{D} \rightarrow \mathbb{D}$ such that $\Delta u(z(d'),d) < \Delta u_{max}$ and $w(z(d'),k) = 0$. In other words, an attack tries to maintain the attacked watermarked object within the usability vicinity of the original non-watermarked one, while making it impossible to recover the watermark. Notation: Let $\mathbb{Z}_w$ be the set of all attacks z for a given w.

Watermark Power:[5] In designing a new metric for the power of a certain watermark we have to take into consideration two main aspects, namely (i) how "rare" is the watermark and (ii) how easy it is to find and apply a successful attack for it. The "rarity" of the watermark is modeled by ϵ as defined above. Estimating real-life raw attack-ability of algorithms is basically intractable, thus we have to assume that a successful attack is always available.

A powerful marking method should result in a watermarked object d', at the "outskirts" of the allowable usability vicinity of the original d, making it hard/impossible for an attacker to directly derive d or the considered vicinity set.

For a given watermarked object d' and associated property w, we define the *power of the watermark* as:

$$power(w,d') = (1-\epsilon) * \frac{1}{P(\exists z \in \mathbb{Z}_w))} * \frac{1}{|V_{d \cap d'}|} \quad (2)$$ [6]

[4] The definition becomes more complex in case of specifying different maximum allowed changes in usability for each given domain. It is simple to derive that case from this one.

[5] The present definition requires more careful attention and many future refinements are envisioned. It is given only as an illustrative example of the new proposed approach and is not to be taken directly to implementation.

[6] The notation $power(w)$ will be used if d' is implied by the context.

Note: $V_{d \cap d'}$ is the intersection of the usability vicinities of d and d'. It defines the target space for any successful attack, effectively modeling the ease of finding a non-watermarked, usable version of d'.

Note: $P(\exists z \in \mathbb{Z}_a)$ defines the probability that a successful attack can be found for a given algorithm. In the following we will consider it 1 (highly likely).

If we consider $P(\exists z \in \mathbb{Z}_a)) = 1$ we have a formula easier to sample and compute:

$$power(w, d') = (1 - \epsilon) * \frac{1}{|V_{d \cap d'}|} \quad \text{(2-2)}$$

The power of a certain watermark is directly related to its convince-ability towards Jared the Judge. The weaker the watermark (higher the false hit probability upper-bound) the less convincing it will be.

Note: It is to be noted that in real life, a certain watermark embedded into an object can be 'viewed' through different property functionals, that is there can be multiple w's that reveal the given base watermark with different ϵ's. This basically corresponds to different methods of watermark detection. The concept of *power* is also distinguishing among them.

Relative Algorithm Power: If given two algorithms $a_1, a_2 \in \mathbb{A}_{\mathbb{D}}$ we say that a_1 is *weaker than* a_2 *with respect to* d if, whenever applied to the same object $d \in \mathbb{D}$ and key $k \in \mathbb{K}$, a_1 returns an associated (property functional,object) pair (w_1, d_1) that is weaker than the one returned by a_2.

Weighted Algorithm Power: Whereas relative algorithm power indeed compares two applications (i.e. with respect to a certain object $d' \in \mathbb{D}$) of the algorithms we need a stronger, broader [7], way of measuring watermarking effectiveness of algorithms.

[7] The required "breadth" derives from the fact that we would like to be able to assert that "in general" one algorithm is better than another.

Let there be a certain distribution $\mathbb{F} = (f_i)_{i \in (1, |\mathbb{D}|)}$ over the objects of $\mathbb{D} = \{d_1, d_2, ..., d_i, ...\}$, with $\sum f_i = 1$ (e.g. f_i could be the probability that a certain object $d_i \in \mathbb{D}$ will be considered for watermarking/attack/malicious use, this can be estimated statistically by normalized counts). Let $a \in \mathbb{A}$ be a watermarking algorithm considered. If we use the notation $a(d_i, k_i) = (d'_i, w'_i)$ we define the *weighted algorithm watermarking power* by the following formula:

$$power = \frac{\sum f_i * power(w'_i, d'_i)}{|\mathbb{D}|} \quad \text{(3)}$$

Note: If we assume that $f_i = f_j \forall i, j$ then the above definition basically converges to the average of the power of all individual applications to all the objects in $\mathbb{D}$.

Main Challenge: Power and Usability. Given a maximum level for difference in usability Δu_{max} and a maximum upper bound on the false positive probability ϵ_{max}, the first challenge of watermarking is to find the most powerful marking algorithm $a \in \mathbb{A}_{\mathbb{D}}$ for a given key $k \in \mathbb{K}$ that still works within the given usability bounds, that is, $\Delta u(d', d) < \Delta u_{max}$ and $P(w(x, k) = 1 | w(d', k) = 1) < \epsilon < \epsilon_{max}$ for all x inside of d's usability vicinity of radius Δu_{max}, where d' is defined by $a(d, k) = (d', w)$.

In other words, the main concern in watermarking lies with keeping the required usability level of the object unchanged or close to its original value, while still featuring enough power. Thus, an appropriate algorithm will try to determine the main usability domains for a particular to-be-watermarked object and then preserve usability in those domains.

3 Scenario

One could argue that, using capacity as a measure of the power of "persuasion" of a certain algorithm works, if, instead of hiding a known text (e.g. "This is the property of Alice"), the algorithm hides a hash of it or some other form of encrypted secret. This - the argument goes - will increase the actual "persuasiveness" of the algorithm and will tightly relate capacity to the convince-ability towards Jared, because, after all, only Alice could have known the encoded secret and the probability that anyone else might know it, is computationally zero and thus secure.

Whereas the above argument makes a good point of showing that hiding a secret (i.e. that looks like random "garbage" to the attacker) is much better than hiding a plain-text message, it misses the point in case of existing trivial attacks on the algorithm as a whole.

In the following we present a simple scenario in which a metric based mainly on hiding capacity cannot predict the weakness of a marking algorithm, whereas the *weighted algorithm watermarking power* metric performs well.

For illustration purposes, consider an algorithm in the space of LSB watermarking algorithms. Those algorithms are known to be weak, because of trivial attacks that can be successfully deployed against them [2] [13] [14].

Let $\mathbb{D}$ be the space of images (e.g. JPEG pictures). Given some normalized image distortion metric in a trivial usability domain (e.g. HVS - Human Visual System), $m : \mathbb{D} \times \mathbb{D} \rightarrow [0, 1]$, lets consider $d \in \mathbb{D}$ and the usability vicinity of d of radius $\Delta u_{max} = m(d, d')$. $d' \in \mathbb{D}$ is obtained from d by altering the LSB subset of d such that $m(d, d')$ is maximal [8].

If we assume (like it is generally understood) that altering LSB information (to the extent given above) is usually tolerable [9], because the given usability vicinity contains $O(2^{|LSB|})$ elements (many), and because $P(w(x, k) = 1 | w(d', k) = 1)$ as defined in (1) cannot be upper bound (i.e. it is 1), the power of any LSB algorithm tends to 0 (weakest) rendering the algorithm (rightfully) unusable in Court.

On the other hand, if *available encoding capacity* (capacity in LSB can be arbitrary large, depending on the encoding method and on the size of the LSB space) would have been a major factor in providing Court confidence, and knowledge about obvious attacks (e.g. zero all LSB info) would not have been available, then the alleged non-zero confidence may have determined its undeserved use in Court proofs.

Although the example is not the most accurate one (defining a "perfect" case is out of the space requirements of this poster) it certainly is relevant intuitively, linking the idea of measuring watermarking algorithm quality to its final goal, ownership proof and creation rights affirmation in Court.

4 Conclusions

We defined main generic watermarking domain issues and presented a new metric aiming at qualitatively measuring watermarking algorithms.

We stressed our metric, *watermarking power*, versus any arbitrary domain-specific metric as being a better formula, essentially linked to the underlying final purpose of any watermarking algorithm. A

[8] We refrained from actually specifying the method for simplicity purposes. In some cases, zero-ing the LSB information in d will achieve a maximal distortion but specifying this method is not important to our point.

[9] That is, Δu_{max}, the usability vicinity radius, is accepted by Jared the Judge in a Court proof.

simple example was given, outlining the main differences.

Further work is required in refining the given metric and determining different data specific applications. Integration with existing domain specific work is another point of future interest.

References

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM Systems Journal*, 35(3–4):313–336, 1996.

[2] Walter Bender, Daniel Gruhl, and Norishige Morimoto. Techniques for data hiding. In *Proc. of SPIE*, volume 2420, pages 40–48, February 1995.

[3] G. Cohen, S. Encheva, and G. Zemor. Copyright protection for digital data. In *Workshop on Watermarking and Copyright Enforcement*, Paris France, 1999.

[4] Ingemar J. Cox and Jean-Paul M. G. Linnartz. Public watermarks and resistance to tampering. In *International Conference on Image Processing (ICIP'97)*, Santa Barbara, California, U.S.A., 26–29 October 1997. IEEE.

[5] Ingemar J. Cox, Matt L. Miller, and A. L. McKellips. Watermarking as communications with side information. *Proceedings of the IEEE (USA)*, 87(7):1127–1141, July 1999.

[6] Scott Craver. On public-key steganography in the presence of an active warden. In *Information Hiding*, pages 355–368, 1998.

[7] Edward J. Delp. Watermarking: Who cares? does it work? In Jana Dittmann, Petra Wohlmacher, Patrick Horster, and Ralf Steinmetz, editors, *Multimedia and Security – Workshop at ACM Multimedia'98*, volume 41 of *GMD Report*, pages 123–137, Bristol, United Kingdom, September 1998. ACM, GMD – Forschungszentrum Informationstechnik GmbH, Darmstadt, Germany.

[8] Bob Ellis. Public policy: New on-line surveys: Digital watermarking. *Computer Graphics*, 33(1):39–39, February 1999.

[9] M. Kobayashi. Digital watermarking: Historical roots. IBM Research Report RT0199, IBM Japan, Tokyo, Japan, April 1997.

[10] Martin Kutter and Fabien A. P. Petitcolas. A fair benchmark for image watermarking systems. In *citeseer.nj.nec.com/article/kutter99fair.html*, pages 226–239.

[11] P. Moulin and J. O'Sullivan. Information-theoretic analysis of information hiding. In *P. Moulin and J. A. O'Sullivan, Information-Theoretic Analysis of Information Hiding*, 1999.

[12] Pierre Moulin, M. K. Mihcak, and Gen-Iu (Alan) Lin. An information–theoretic model for image watermarking and data hiding. In *(manuscript)*, 2000.

[13] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Attacks on copyright marking systems. In David Aucsmith, editor, *Information Hiding: Second International Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 218–238, Portland, Oregon, U.S.A., 1998. Springer-Verlag.

[14] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding - a survey. *Proceedings of the IEEE*, 87(7):1062–1078, July 1999. Special issue on protection of multimedia content.

[15] B. Pfitzmann. Information hiding terminology. In Ross Anderson, editor, *Information hiding: first international workshop, Cambridge, U.K., May 30–June 1, 1996: proceedings*, volume 1174 of *Lecture Notes in Computer Science*, pages 347–350, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1996. Springer-Verlag.

[16] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, 1949.

[17] Radu Sion. Analog output-stage fingerprinting in DMAT/SDMI. In *Proceedings of WM02, CERIAS TR 2001-53*, 2002.

[18] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. On watermarking numeric sets. In *Proceedings of IWDW 2002, Lecture Notes in Computer Science, CERIAS TR 2001-60*. Springer-Verlag, 2002.

[19] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. On watermarking semistructures. In *(submission for review), CERIAS TR 2001-54*, 2002.

[20] Radu Sion, Mikhail Attalah, and Sunil Prabhakar. Key pre-commitment in watermarking. In *(submission for review), CERIAS TR 2002*, 2002.

[21] G. Voyatzis, N. Nikolaidis, and I. Pitas. Digital watermarking: an overview. In S. Theodoridis et al., editors, *Signal processing IX, theories and applications: proceedings of Eusipco-98, Ninth European Signal Processing Conference, Rhodes, Greece, 8–11 September 1998*, pages 9–12, Patras, Greece, 1998. Typorama Editions.

[22] Jian Zhao and Eckhard Koch. A generic digital watermarking model. *Computers and Graphics*, 22(4):397–403, August 1998.

[23] Jian Zhao, Eckhard Koch, Joe O'Ruanaidh, and Minerva M. Yeung. Digital watermarking: what will it do for me? and what it won't! In ACM, editor, *SIGGRAPH 99. Proceedings of the 1999 SIGGRAPH annual conference: Conference abstracts and applications*, Computer Graphics, pages 153–155, New York, NY 10036, USA, 1999. ACM Press.

Radu Sion (sion@cs.purdue.edu) is a PhD student in Computer Sciences at Purdue University. He received his MSc from Purdue University in 2000 and from Politehnica University of Bucharest, Romania, in 1999 and a BSc from Politehnica University of Bucharest in 1998. His interests include Steganography and Information Hiding, applications of Timing Patterns modeled using Neural Networks in intrusion detection (e.g. ”Network Breath”), Peer to Peer Computing, content based delivery networks, Distributed Systems Security. His PhD disertation embraces an original idea in the broader area of steganography and watermarking, namely information hiding in semi-structures and aggregates with applications ranging from numeric set watermarking to XML tamperproofing.

Prof. **Mikhail Atallah** is a professor in the Computer Sciences Department, Purdue University. He is a Fellow of the IEEE, and serves or has served on the editorial boards of *SIAM J. on Computing, J. of Parallel and Distributed Computing, Information Processing Letters, Computational Geometry: Theory & Applications, Int. J. of Computational Geometry & Applications, Parallel Processing Letters, Methods of Logic in Computer Science*. He was Guest Editor for a Special Issue of *Algorithmica* on Computational Geometry, has served as Editor of the *Handbook of Parallel and Distributed Computing* (McGraw-Hill), as Editorial Advisor for the *Handbook of Computer Science and Engineering* (CRC Press), and serves as Editor in Chief for the *Handbook of Algorithms and Theory of Computation* (CRC Press). He has also served on many conference program committees, and state and federal panels. His research interests include the design and analysis of algorithms, in particular for the application areas of computer security and computational geometry.

Prof. **Sunil Prabhakar**’s research focuses on issues in large-scale, distributed applications such as multimedia databases, data warehouses, and digital libraries. The efficient execution of I/O is a critical problem for these applications. He is currently developing techniques that improve I/O performance for traditional and multimedia databases. He has developed declustering algorithms for multidimensional data that result in increased parallel I/O scheduling algorithms for robotic removable media libraries. Dr. Prabhakar’s interest also lies in the design and development of digital libraries for the management and study of scientific research data.