Energy Analysis of Multimedia Watermarking on Mobile Handheld Devices

Arun Kejariwal[†] Sumit Gupta[‡] Alexandru Nicolau[†] Nikil Dutt[†] Rajesh Gupta[§]

[†]School of Information and Computer Science
University of California at Irvine
arun_kejariwal@ieee.org
{nicolau, dutt}@cecs.uci.edu

[‡]Tensilica Inc.
sumitg@ieee.org

[§]Dept. of Computer Science and Engineering
University of California at San Diego
gupta@cs.ucsd.edu

Abstract

The recent growth of internet-enabled handheld devices has increased the need for the protection of digital media and intellectual property rights. One way to ensure this protection is via digital watermarking, a process that embeds an imperceptible signature or watermark in a digital file containing audio, image, text or video data. The signature is later used to authenticate the data file and for tamper detection. Effective watermarking can be computationally expensive and place additional drain on the available energy in handheld devices. We present a detailed analysis of the energy requirements and performance of a range of prevalent and proposed algorithms for signature embedding, extraction, and authentication for images. We also discuss the security versus energy trade-offs of these algorithms and study the impact of image resolution on energy and performance.

1 Introduction

The internet is being increasingly used for the distribution of digital multimedia content. The inherent flexibility of the internet enables users and commercial entities to create, distribute, store, peruse, modify, and trade digital content in various forms such as text documents, databases, e-books, images, audio, video, computer software, and games. Such flexibility gives rise to concerns about the protection and enforcement of intellectual property (IP) rights of the digital content provider involved in the transaction. In addition, unauthorized replication and manipulation of digital content is relatively trivial and can be done using inexpensive tools, unlike traditional analog multimedia content.

The use of watermarks has been proposed as a solution to protect the intellectual property of digital products. *Digital watermarking* [1, 2, 3, 4] is the process of embedding a secret code or identification tag into a digital media file so that it is hidden from view, but can be extracted on demand to verify the authenticity of the media file. Watermarks are embedded in applications in the form of additional information like identifiers of the owner, recipient, and/or distributor, transaction dates, serial numbers, et cetera.

Handheld devices such as PDAs (personal digital assistants) and cell phones are increasingly being used for exchanging, viewing, and listening to digital content. Mobile devices have limited computational capability owing to their small form factor. Furthermore, these devices have a limited energy budget (battery life is limited). Performing digital watermarking on mobile handheld devices places an additional computational and power burden on these devices.

In this paper, we analyze the energy requirements and execution time for image watermarking algorithms on mobile devices. We also examine the effect of increased security (better watermarks) and image resolution on energy and execution time. We believe that such characterization is critical for addressing energy efficient watermarking for battery-constrained mobile handheld devices. In addition, the analysis helps to determine a set of "profitable" (from performance/energy perspective) computations for proxy-based offloading [5]. To the best of our knowledge, this is the first detailed characterization of the energy requirements of watermarking algorithms.

The rest of the paper is organized as follows. In Section 2, we present an overview of signature generation and image watermarking. In Section 3, we discuss the architecture of our target system. Section 4 presents the experimental setup and results. In the next section, we present related work. Finally, Section 6 summarizes this work with directions for future work.

2 Watermarking

Digital watermarking has been proposed as a technique for protecting intellectual property in the digital world. This includes preventing unauthorized copying and validating authenticity of data, whether it be audio, textual, visual, or video. Watermarking can also be used to demonstrate proof of ownership, to detect if a digital file has been changed or tampered with, to embed additional data (such as author information), and to hide data. Watermarking has also been proposed for software mobile agents to detect if a host has tampered with the mobile agent code [4].

The tasks involved in watermarking for images are shown in Figures 1(a) and (b). There are three basic tasks in the watermarking process (with respect to an image):

- *Embedding*: Refers to invisibly implanting a watermark into an image. The watermark itself is generated using a signature generation process. A residual

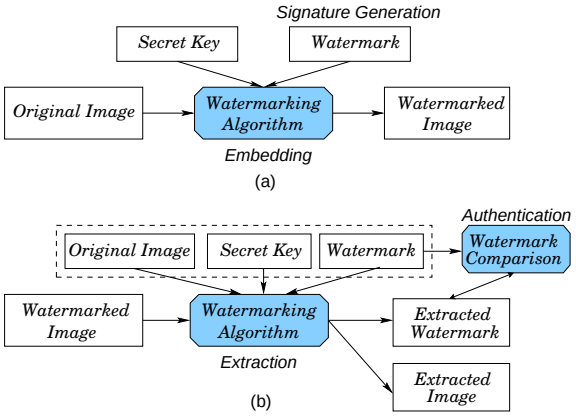


Figure 1. Watermarking process: (a) Watermark Generation and Embedding, (b) Watermark Extraction and Authentication.

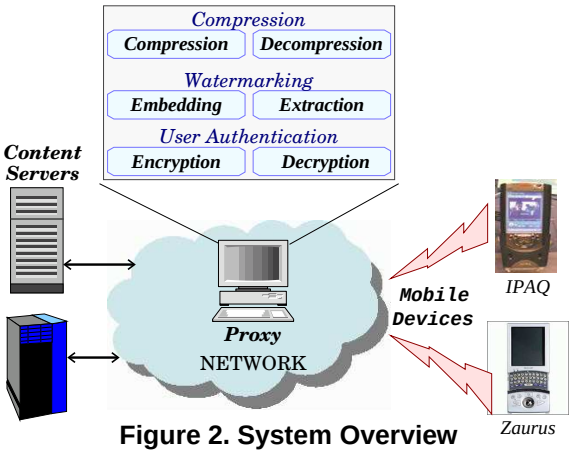


Figure 2. System Overview

mask is used for the embedding procedure. The mask modifies the neighboring relationship between the coefficients of the transformed image. The quantity of modification is used as a measure of invisibility and robustness of the embedding procedure. The watermark embedding algorithm may use a secret key to determine the way in which the watermark is embedded into the image [2].

- *Detection and Extraction*: Refers to detecting whether an image has a watermark and extracting this watermark from the image. Watermarking algorithms require either the original image [6] or the original watermark [7] or the secret key [2], or any combination thereof, to extract the watermark.
- *Authentication*: Refers to comparing the extracted watermark with the original watermark. In general, a threshold is defined for the comparison. If the differences (if any) are more than the threshold the image is declared tampered.

Watermarks (or digital signatures) are used to detect unauthorized modifications of data and for ownership authentication. In addition, the recipient of signed data can use a digital signature in proving to a third party that the signature was in fact generated by the signer of the data. This is known as *non-repudiation* since the signer of data cannot, at a later time, repudiate the signature.

2.1 Image Watermarking Techniques

A simple approach for embedding data into images is to set the least significant bit (LSB) of some pixels to zero. Data is then embedded into the image by assigning 1s to the LSBs in a specific manner (known only to the owner). This method satisfies the perceptual transparency property, since only the least significant bit of an 8-bit value is altered.

The quantization tables used in lossy compression techniques take advantage of the fact that the human visual system is less sensitive to quantization noise at higher frequencies. That is, the human eye is more sensitive to noise in the lower frequency range than in the higher frequency range, while the energy of most of natural images is concentrated on the lower frequency range. Therefore, in order to invisibly embed a watermark and survive lossy data compression, a reasonable trade-off is to embed the watermark into the middle-frequency range of the image [8].

In DCT-based watermarking, the original image is divided into 8×8 blocks of pixels, and the 2-D DCT (discrete cosine transform) is applied independently to each block. The watermark is then embedded into the image by modifying the relationship of the neighboring blocks of the DCT coefficients that are in the middle-frequency range in the original image [8].

Each RGB (red-green-blue) component in color images may have a different resolution. Wavelet-based watermarking techniques exploits this by creating a multi-resolution representation in which each RGB component is represented in a different frequency band. This, however, leads to computationally expensive searches for coefficients across sub-bands [9] while embedding and extracting the watermark. The watermark is adaptively weighted in to different sub-bands to achieve robustness as well as imperceptible watermarks.

3 System Architecture

An overview of the architecture of our target system is shown in Figure 2. This system consists of three components: mobile devices, proxy servers and servers. A *mobile device* refers to any type of networked resource; it could be a handheld (personal digital assistant or PDA), a printer or a wireless security camera et cetera. *Servers* store the multimedia and database content and stream data (say images) to a client as per requests. All communication between the mobile devices and the servers are relayed through the *proxy* servers. Proxy servers are powerful servers that can, among other things, compress/decompress images, transcode video in real-time, access/provide directory services, and provide services based on a rule base for specific devices. Mobile devices thus actually negotiate with proxy servers for security, quality of service and content delivery. The proxy servers in turn request the content servers for the image/video/data stream as per user requirements.

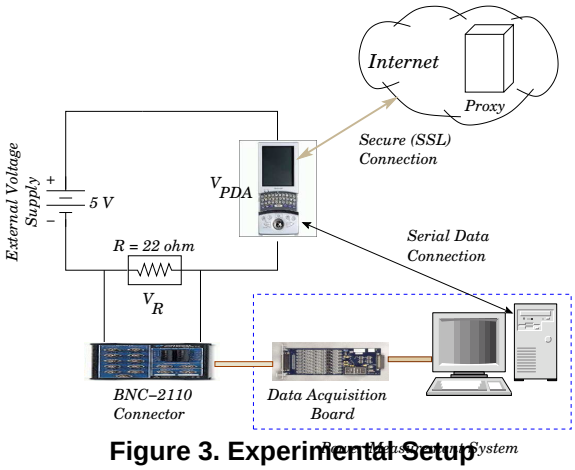


Figure 3. Experimental Setup

4 Experimental Setup and Results

In this section, we present experimental results for performance and energy consumption of a suite of 11 watermarking algorithms on a handheld.

4.1 Setup

Our experimental setup is shown in Figure 3. All our measurements were made using a Sharp Zaurus PDA with an Intel 400MHz XScale® processor with 64MB ROM and 32MB SDRAM. We used a Belkin 802.11b wireless USB network adaptor on the Zaurus for communication. We removed the batteries from the Zaurus and placed a resistor in series with the power supply, as shown in Figure 3. We used a National Instruments PCI DAQ (data acquisition) board to sample the voltage drop across the resistor (to calculate current) at 1000 samples/second. We calculated the instantaneous power consumption corresponding to each sample and the total energy using the following equations :

$$\mathcal{P}_{Inst} = \frac{V_R}{R} \times V_{PDA} \quad (1)$$

$$E = \sum \mathcal{P}_{Inst} \times T \quad (2)$$

where V_R is the instantaneous voltage drop across the resistor (in volts) with resistance R ohms and V_{PDA} is the voltage across the Zaurus PDA (or the supply voltage), and T is the sampling period (T = 1/1000 since sample rate is 1000 samples/second). Thus, as per Equation 2 above, energy E is the sum of all the instantaneous power samples for the duration of the execution of the application multiplied by the sampling period T. We calculate *average power* as the ratio of total energy over total execution time.

We performed experiments on 11 image watermarking algorithms, referred to as Bruyndonckx, Corvi, Cox, Dugad, Fridrich, Kim, Koch, Wang, Xia, Xie and Zhu,[1] whose source code is available from [11]. The source code was cross-compiled using *arm-gcc* for Zaurus. This suite of watermarking algorithms is representative of a range of image watermarking techniques and as our results show, have very different performance and energy consumption.

[1]For detailed description of each algorithm the reader is referred to [10].

Algorithm	Energy (J)	Execution time (s)	Avg. Power (W = J/s)
Bruyndonckx	1.47	13.46	0.11
Corvi	83.2	136.15	0.61
Cox	126	115.23	1.1
Dugad	68.7	136.64	0.50
Fridrich	196	171	1.15
Kim	73.5	140.81	0.52
Koch	2.19	12.64	0.17
Wang	85.8	140.2	0.61
Xia	90	133.82	0.67
Xie	154.8	147.07	1.05
Zhu	163.3	143.74	1.14

Table 1. *Watermark Embedding: Energy, power and execution time analysis*

4.2 Watermark Embedding

Table 1 lists the energy usage, average power (energy/exe-cution time), and execution time for watermark embedding by the various watermarking algorithms when they are executed on the handheld.

From Table 1 we note that Bruyndonckx's and Koch's algorithms are the most energy and performance efficient. This can be attributed to their low security levels as compared to other algorithms. For example, Bruyndonckx embeds most of the signature bits in a few blocks in the image, thus, enabling quick, low computation searches. This speeds up the embedding process and leads to low energy consumption. However, such localized embedding strategies are highly susceptible to watermark attacks [12]. Similarly, Koch embeds signature bits by modifying coefficient[2] relationships using mean square error to minimize error. As compared to other frequency domain watermark embedding techniques, Koch's algorithm offers a very low security level owing to the simple embedding criteria. In contrast, Xie and Zhu employ wavelet-based watermarking techniques; such sub-band level embedding techniques are more secure than other techniques. However, calculating wavelet and inverse-wavelet transforms is computationally expensive and thus, also power hungry.

4.3 Watermark Extraction

We present the energy, power, and execution time analysis of watermark extraction in Table 2. Again, we observe that Bruyndonckx's and Koch's watermarking algorithms consume much less power (up to two orders of magnitude) than others. In the case of Bruyndonckx's extraction procedure, as stated in the previous section, this is due to the locality of the embedded watermark. Also, the extraction process is restricted to the spatial domain, i.e., the search for the embedded watermark is performed in the *blocks* [3] of the image. Whereas in Koch's extraction procedure, the signature is extracted on the basis of the modified relationship of the DCT coefficients. It merely involves extracting the

[2]The coefficients are obtained by carrying out a 8×8 DCT transform on the input image.

[3]A *block* in an image corresponds to a 8×8 block used in DCT (discrete-cosine-transform).

middle frequency coefficient of a 8×8 block. Thus, Koch's algorithm also requires little energy and executes fast.

Algorithm	Energy (J)	Execution time (s)	Avg. Power (W = J/s)
Bruyndonckx	0.22	0.28	0.79
Corvi	70.3	150.77	0.47
Cox	121	128.02	0.95
Dugad	38.4	79	0.49
Fridrich	191	173.6	1.1
Kim	91.3	166.57	0.55
Koch	0.61	1.0	0.61
Wang	88	147.9	0.59
Xia	82.7	144.51	0.57
Xie	74.06	73.88	1.0
Zhu	158.8	137.38	1.16

Table 2. *Watermark Extraction: Energy, power and execution time analysis*

On the other hand, the watermark extraction in Corvi and Kim is done in the wavelet domain. In this type of watermarking algorithms, we first have to determine the coefficients by calculating the wavelet transform and evaluate the *decomposition levels* and their sub-bands. Then these sub-bands are used for watermark extraction. This involves complex computations and hence, higher energy usage. Of all the watermarking algorithms presented in Table 2, Fridrich has the highest energy consumption and execution time because it calculates the mean and the variance of all the pixel values in the image and performs band-wise correlation and image normalization.

We find that in some cases watermark extraction is more expensive than watermark embedding. Wavelet-transform based watermarking techniques such as Fridrich's, compute the transform once on the input image while embedding the watermark. The coefficients are first normalized (this requires mean and derivation calculation) and then embedded *imperceptibly* in the medium frequency range. On the other hand, during extraction, the transform is carried out on both the input image and the output image and the corresponding coefficients are normalized. The correlation between the normalized coefficients of the input and output is used as a measure of the fidelity of the watermarked image. The overhead of computing band-wise correlation and image normalization accounts for the higher energy consumption.

4.4 Watermark Authentication

In Table 3, we list the energy, power and execution time for watermark authentication. This task is computationally inexpensive, since it involves a simple comparison of the extracted watermark and the original watermark. From Table 3, we find that *blind watermarking* techniques such as Bruyndonckx's in which the mark is verified without using the original image, are more power efficient as they forgo fidelity checks on the extracted image. On the other hand, *fragile* watermarking techniques such as Fridrich's in which watermarks are designed to be distorted or "broken" under the slightest changes to the image, employ correlation measures between the original image and extracted image for image verification.

Algorithm	Energy (J)	Avg. Power (W)	Execution time (s)
Bruyndonckx	0.02	0.59	0.034
Corvi	0.10	0.73	0.138
Cox	0.05	1.35	0.037
Dugad	0.03	0.97	0.031
Fridrich	0.18	1.36	0.132
Kim	0.10	0.76	0.131
Koch	0.04	1.25	0.032
Wang	0.08	1.36	0.059
Xia	0.08	1.4	0.057
Xie	0.04	1.0	0.039
Zhu	0.06	1.2	0.05

Table 3. *Watermark Authentication: Energy, power, and execution time analysis*

4.5 Impact of Different Security Levels

Intuitively, it is obvious that the more secure our watermarking scheme, the more computation it will require and the more energy it will consume. In this section, we quantify this intuition by varying the security level for the watermarking algorithms used in our experiments.

In Figures 4(a) and 4(b), we present the impact of increasing the security level, i.e., the number of bits used to generate the watermark, on execution time and energy requirements of watermark embedding and watermark extraction for the 11 watermarking algorithms[4]. For nearly all the watermarking algorithms, we find that the energy consumption increases almost linearly as the watermark length is increased.

From Figures 4(a) and 4(b), we find that increasing the length of the watermark has little performance and energy penalty on local watermark embedding techniques such as Bruyndonckx's. Similarly, Koch's extraction process incurs minimum energy overhead with lower security levels (see discussion in previous three sections). However global embedding techniques such as Fridrich's, where the signature is distributed across a large number of blocks, incur significant penalty as the length of the signature is increased.

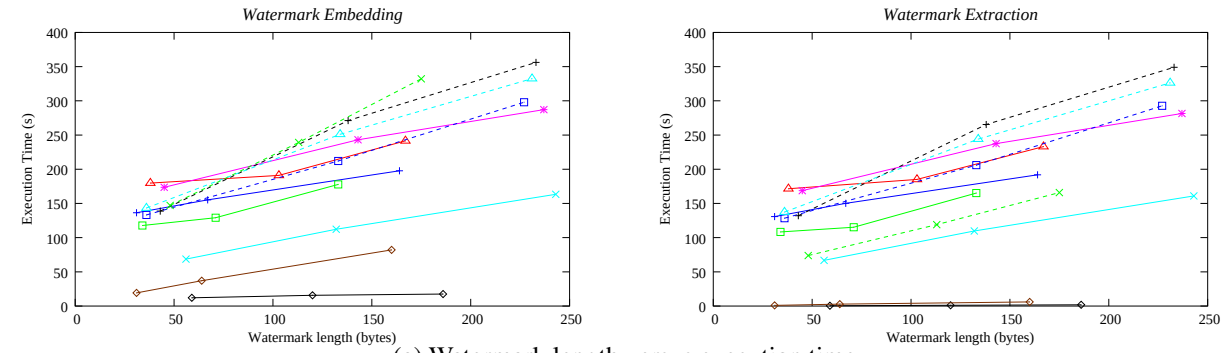
Xie and Zhu's algorithms embed watermarks in the frequency domain. Increased watermark length increases the computation of the correlation between frequency coefficients of the original image and the watermarked image. This correlation computation is a very complex operation and thus, places a heavy power burden on the handheld.

This study of different security levels is useful because frequently the choice of security level must be made in conjunction with the application requirements such as real-time constraints, QoS (quality-of-service), and residual battery energy in the handheld device.

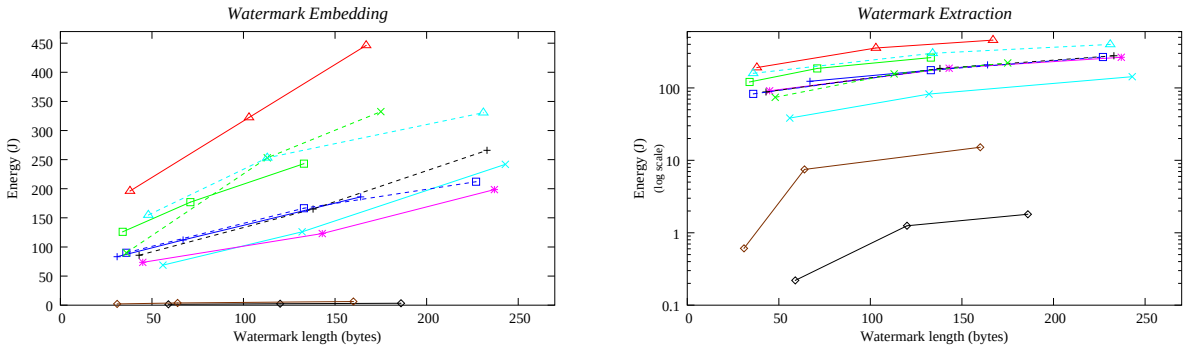
4.6 Impact of Image Resolution

In the last set of experiments, we vary the resolution of the image being watermarked and study its effects on energy and execution time. The impact of varying the image resolution on watermark embedding and extraction are

[4]In the rest of the experiments in this paper, we used a watermark length of 32 bytes.



(a) Watermark length versus execution time



(b) Watermark length versus energy consumption

Figure 4. *Effects of increasing watermark security (length) on execution time and energy consumption.*

presented in Figures 5(a) and (b). From these figures, we observe that Fridrich's algorithm suffers maximum performance and energy penalty with increase in image resolution. As the image resolution increases, from a black and white (2 bit) image to gray-scale (8 bit) image to a color image (24 bit)[5], the number of frequency coefficients per pixel increase, even though the total number of pixels remain the same. The larger number of coefficients lead to more computations (calculation of mean, band-wise correlation, and image normalization). Thus, spatial domain filters are "immune" to image resolution variance, whereas the performance of frequency domain filters are susceptible to image resolution.

High resolution images have higher quality, but might require more energy to download (communication energy) and to render (computational energy). On the other hand, from a security viewpoint, high resolution images provide a larger spectrum for non-uniform embedding of the watermark along each plane. [6] However, this too leads to higher energy consumption, since the search space for embedding and extracting the watermark increases.

5 Related Work

The term *watermark* has traditionally referred to an almost imperceptible imprint on paper that marks the authenticity of the document. Digital watermarking extends this idea to digital content [1, 2, 3, 13, 4]. Watermarking like steganography seeks to hide information inside another object, but should be resilient to intentional or unintentional manipulations and resistant to watermark attacks [2].

The use of proxies as agents that can connect to a range of heterogeneous clients is a well-established practice [14, 15]. Several approaches have been proposed for securing the connection between the proxy and mobile devices that they serve [16, 17]. Also, moving computationally expensive tasks to proxies has been discussed in the past [18] and recently, is used in the context of establishing secure connections [15, 19]. However, in most of these works, task mapping is driven solely by performance, ignoring power constraints that are critical for handheld mobile devices [20, 21]. Recently, Potlapally et. al [22] analyzed the energy requirements of a wide range of cryptographic algorithms used as building blocks in security protocols such as SSL (Secure Sockets Layer).

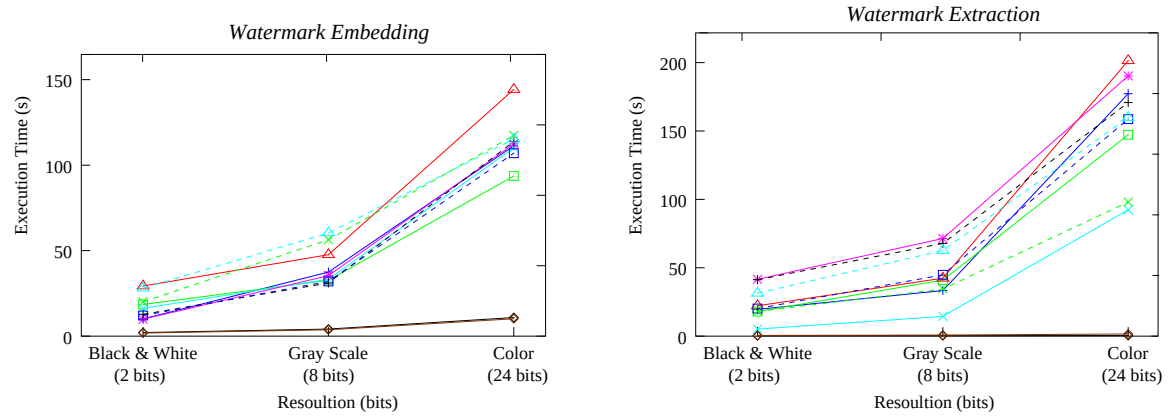
Our work focuses on another computationally expensive task that is becoming interesting in the context of mobile devices – that of protecting digital content copyrights using approaches such as watermarking.

6 Conclusion

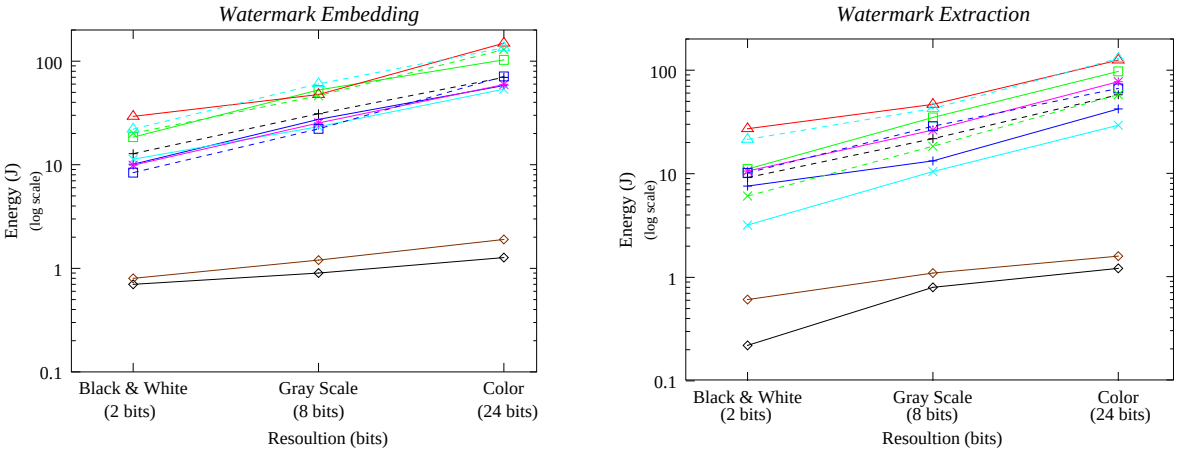
We presented an extensive analysis of the energy and computation requirements of a range of image watermarking algorithms – for each stage of the watermarking process.

[5] Each color component RGB is assigned 8 bits.

[6] Each RGB component in an image forms a plane in the image.



(a) Impact of image resolution on execution time.



(b) Impact of image resolution on energy

Figure 5. *Effects of increasing Image Resolution on execution time and energy consumption.*

We analyzed the effects of parameters such as watermark security level on performance and energy consumption. We gave some insight into the relation between the complexity and security level of the watermarking algorithm and the energy consumed to run the algorithm. We also found that watermark extraction requires more energy than watermark embedding since the former has to search the entire image for a watermark. In future work, we plan to determine task or function level energy consumption of the watermarking algorithms, so that we can develop a global task-based partitioning strategy that minimizes power with a heterogeneous set of authentication and authorization requirements.

References

[1] K. Tanaka, Y. Nakamura, and K. Matsui. Embedding secret information into a dithered multilevel image. In *IEEE Military Communications Conference*, pages 216–220, 1990.

[2] Fernando Pérez-González and Juan R. Hernández. A tutorial on digital watermarking. In *IEEE Annual Carnahan Conference on Security Technology*, 1999.

[3] L. Qiao. *Multimedia Security and Copyright Protection*. PhD thesis, Dept. of Computer Science, University of Illinois at Urbana-Champaign, 1998.

[4] O. Esparza, M. Fernandez, and M. Soriano. Protecting mobile agents by using traceability techniques. In *International Conference on Information Technology*, 2003.

[5] A. Kejariwal, S. Gupta, A. Nicolau, N. Dutt, and R. Gupta. Proxy-based task partitioning of watermarking algorithms for reducing energy consumption in mobile devices. In *Proceedings of the 41st Annual Conference on Design automation*, pages 556–561, San Diego, CA, 2004.

[6] O. Bruyndonckx, J.-J Quisquater, and B. Macq. Spatial method for copyright labeling of digital images. In *Proceedings of IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, Greece*, pages 456–459, June 1995.

[7] M. Corvi and G. Nicchiotti. Wavelet-based image watermarking for copyright protection. In *Scandinavian Conference on Image Analysis, Lappeenranta, Finland*, June 1997.

[8] C.T. Hsu and J.L. Wu. Hidden signatures in images. In *Proceedings of IEEE International Conference on Image Procesing*, pages 223–226, Sep 1996.

[9] R. Dugad, K. Ratakonda, and N. Ahuja. A new wavelet-based scheme for watermarking images. In *Proceedings of International Conference Image Processing*, pages 357–372, Oct 1998.

[10] P. Meerwald. Digital image watermarking in the wavelet transform domain. Master's thesis, Department of Scientific Computing, University of Salzburg, Austria, Jan 2001.

[11] Source code distribution of several watermarking algorithms. `http://www.cosy.sbg.ac.at/~pmeerw/Watermarking/source/`.

[12] S. Voloshynovskiy, S. Pereira, and T. Pun. Watermark attacks. In *Erlangen Watermarking Workshop*, Oct 1999.

[13] Proceedings of the IEEE, Special Issue on "Identification and protection of multimedia information". July 1999.

[14] H. C. Rao, D. Chang, Y. Chen, and M. Chen. iMobile: a proxy-based platform for mobile services. In *Wireless Mobile Internet*, pages 3–10, Rome, Italy, July 2001.

[15] J. G. Steiner, B. C. Neumann, and J. I. Schiller. Kerberos : An authentication service for open network systems. In *Proceedings of Winter USENIX Conference*, pages 191–201, 1988.

[16] B. C. Neuman. Proxy-based authorization and accounting for distributed systems. In *International Conference on Distributed Computing Systems*, pages 283–291, May 1993.

[17] A. Fox and S. D. Gribble. Security on the move: Indirect authentication using kerberos. In *Mobile Computing and Networking*, pages 155–164, White Plains, NY, Nov 1996.

[18] B. Zenel. A proxy based filtering mechanism for the mobile environment. Technical Report CUCS-0-95, Computer Science Department, Columbia University, 1995.

[19] M. Burnside, D. Clarke, T. Mills, A. Maywah, S. Devadas, and R. Rivest. Proxy-based security protocols in networked mobile devices. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 265–272, Madrid, Spain, 2002.

[20] M. Jakobsson and D. Pontcheval. Mutual authentication for low-power mobile devices. In *Proceedings of Financial Cryptography*. Springer-Verlag, 2001.

[21] E. Shih, V. Bahl, and M. Sinclair. Reducing energy consumption of wireless, mobile devices using a secondary low-power channel, March 2003.

[22] N. R. Potlapally, S. R., A. Raghunathan, and N. K. Jha. Analyzing the energy consumption of security protocols. In *Proceedings of the 2003 international symposium on Low power electronics and design*, pages 30–35, Seoul, Korea, 2003.