Energy Analysis of Multimedia Watermarking on Mobile Handheld Devices

Arun Kejariwal[‡] Sumit Gupta[‡]

Alexandru Nicolau[‡] Nikil Dutt[‡] Rajesh Gupta[§]

CECS

Technical Report #03-38

November 2003

Center for Embedded Computer Systems

[‡] Dept. of Information and Computer Science
University of California at Irvine
{sumitg, nicolau, dutt}@cecs.uci.edu

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

http://www.cecs.uci.edu

Abstract

Digital watermarking is a process that embeds an imperceptible signature or watermark in a digital file containing audio, image, text or video data. The watermark is later used to authenticate the data file and for tamper detection. It is particularly valuable in the use and exchange of digital media such as audio and video on emerging handheld devices. However, watermarking is computationally expensive and adds to the drain of the available energy in handheld devices. We present an approach in which we partition the watermarking embedding and extraction algorithms and migrate some tasks to a *proxy* server. This leads to a lower energy consumption on the handheld without compromising the security of the watermarking process. Our results show that executing watermarking partitioned between the proxy and the handheld reduces the total energy consumed by 80% over running it only on the handheld and improves performance by over two orders of magnitude.

Contents

List of Figures

1 Introduction

The increasing computational capability and availability of broadband in emerging handheld devices have made them true endpoints of the internet. They enable users to download and exchange a wide variety of media such as e-books, images et cetera. However, practical use of these devices requires effective protection of IP rights associated with such media.

Digital watermarking [1, 2, 3, 4] has been proposed as a technique for protecting intellectual property of digital data. It is the process of embedding a signature/watermark 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. The watermark can be a binary data, a logo or a seed value to a pseudo-random number generator to produce a sequence of numbers with a certain distribution (e.g., Gaussian or uniform).

In mobile devices, watermarking can be used to combat fraudulent use of wireless voice communications, authenticating the identity of cell phones and transmission stations, and securing the delivery of music and other audio content [5]. Bauerle et al. suggest watermarking images taken by the inbuilt camera in a mobile device (cell phone, PDA) or the GPS coordinates of the user before sending it to the server over a non-secure wireless channel [6]. Products like Digimarc ImageBridge [5], Hitachi's 32-bit RISC processor SH-Mobile (SuperH Mobile Application Processor) chips [7] on cellular phones allow communicating copyrights in digital images and tracking those images on the Internet. Similarly, RealOne Player [8] for mobile devices like the Nokia 9210/9290 and the Compaq iPAQ Pocket PC, provides support to play watermarked streaming RealAudio and RealVideo files live and on-demand. Similarly, security concerns in mobile e-commerce have been discussed in [9]. Watermarking bears a large potential in securing such applications, for example, customer authentication in service delivery and customer support.

Handheld devices such as PDAs (personal digital assistants) and cell phones have a limited battery life that is directly affected by the amount of computational burden placed by the application. Digital watermarking tasks place an additional burden on the available energy in these devices.

In this paper, we propose a task partitioning scheme for wavelet-based watermarking applications in which computationally expensive portions of the watermarking are offloaded to a proxy server. The proxy server acts as an agent between the content server and the handheld device and is used for various other tasks such as data transcoding, load management, et cetera. We show how our partitioning scheme can be used to reduce energy consumption associated with watermarking on the handheld without compromising the security of the watermarking process.

The rest of the paper is organized as follows. In the next section, we present related work. In Section 4, we present an overview of the watermarking process. Next we discuss wavelet-based watermarking and then the architecture of our target system. In Section 6, we present our partitioning approaches. Finally, in Section 7 we present our experimental setup and results and conclude our work in Section 8.

2 Related Work

This work builds upon ongoing advances in digital watermarking and proxy-based middleware services. The term *watermark* traditionally refers 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, 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 [10, 11]. Several approaches have been proposed for securing the connection between the proxy and mobile devices that they serve [12, 13]. Also, moving computationally expensive tasks to proxies has been discussed in the past [14] and recently, is used in the context of establishing secure connections [11, 15]. However, in most of these works, task mapping is driven solely by performance, ignoring power constraints that are critical for handheld mobile devices. Recently, Potlapally et. al [16] analyzed the energy requirements of a wide range of cryptographic

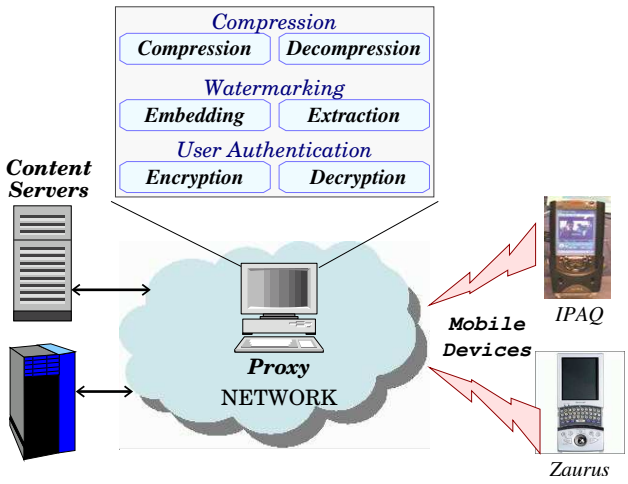

Figure 1: Architecture of Target System

algorithms used as building blocks in security protocols such as SSL (Secure Sockets Layer).

3 System Architecture

Figure 1 shows our implementation of a watermarking system in which multimedia content is streamed to a handheld device via a proxy server. This system consists of three components: mobile devices, proxy servers and content servers. A *mobile or handheld device* refers to any type of networked resource; it could be a handheld (personal digital assistant or PDA), a gaming device or a wireless security camera. *Content servers* store 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. Note that mobile devices may also create and send data through proxy servers to other mobile devices

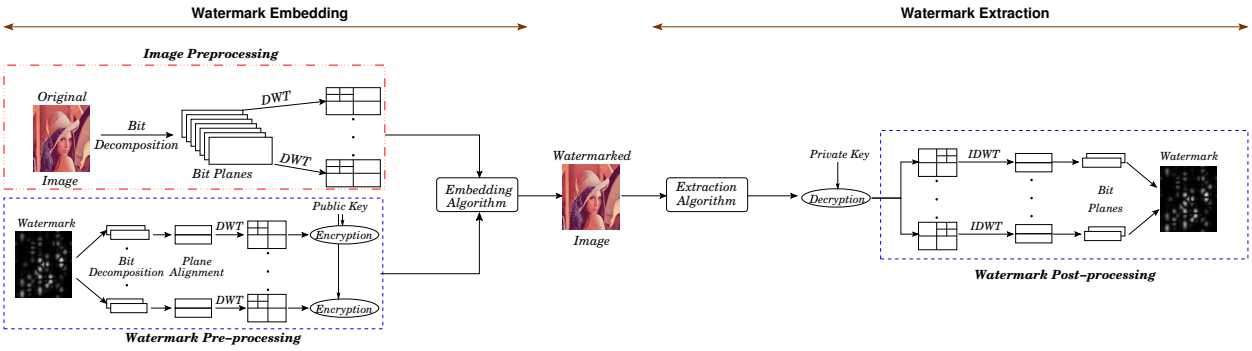

Figure 2: Image watermark embedding and extraction process (public key used for encryption, whereas private key is used for decryption.

in the network.

In the rest of the paper, we discuss techniques that exploit the capabilities of proxy servers to dynamically offload watermarking tasks from handheld devices under constraints on the bandwidth available to the client devices.

4 Watermarking

Figure 3 shows the general process of watermarking image data. The original image (host image) is modified using a signature to create the watermarked image. In this process, some error or distortion is introduced. To ensure transparency of the embedded data, the amount of image distortion due to the watermark embedding process has to be small. The are three basic tasks in the watermarking process with respect to an image as shown in Figure 3(a-b). These are:

- *Embedding*: A watermark is embedded either in the spatial domain (by modifying the relationship of a pixel with its neighboring pixels) or in the frequency domain (by modifying the DCT/DWT[1] coefficients) such that the watermark is imperceptible in the watermarked image.

[1]DCT = Discrete cosine transform, DWT = Discrete wavelet transform

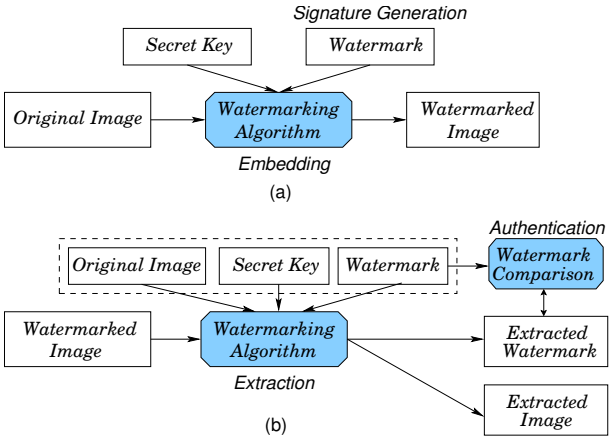

Figure 3: Watermarking process: (a) Watermark Generation and Embedding, (b) Watermark Extraction and Authentication.

Embedding a watermark into an image 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. The watermark is extracted from the image by applying a process that is almost the inverse of the embedding process.
- *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, then the image is declared as tampered.

5 Wavelet-based Watermarking

Wavelet-based watermarking is one of the most popular approaches due to its robustness against malicious attacks [17, 18]. Wavelet-based image watermark embedding consists of three phases – (a) watermark preprocessing, (b) image preprocessing, and (c) watermark embedding, as shown in Figure 2. First, each bit in each pixel of both the image and the watermark is assigned to a bit plane. There are 8 *bit plane* corresponding to the gray level resolution of the image/watermark.

Then DWT (discrete-wavelet transform) coefficients are obtained for each bit plane by carrying out DWT on a plane-by-plane basis. The DWT coefficients of the watermark are encrypted using a *public key*[2]. The watermark embedding algorithm then uses the coefficients of the original image and those of the encrypted watermark to generate the watermarked image.

A similar reverse process is used for watermark extraction and authentication. First the encrypted coefficients for the image and the watermark are extracted from the image. Then a secret *private key* is used to decrypt the coefficients of the watermark and a inverse DWT is applied and so on, till the original image and the watermark are obtained.

5.1 Discrete Wavelet Transform (DWT)

Discrete wavelet transform (DWT) [20] uses filters with different cutoff frequencies to analyze a signal (image/video in our case) at different resolutions[3]. The signal is passed through a series of high-pass filters, also known as *wavelet* functions, to analyze the high frequencies and it is passed through a series of low-pass filters, also known as *scaling* functions, to analyze the low frequencies.

After filtering, half of the samples can be eliminated according to the Nyquist rule, since the signal now has a highest frequency of $p/2$ radians instead of p. The signal can therefore be sub-sampled by 2, simply by discarding every other sample. This constitutes one level of *decomposition*. Thus, decomposition halves the time resolution (half the number of samples) and doubles the frequency resolution (half the span in the frequency band). The above procedure, also known as the *sub-band coding*, is repeated for further decomposition. The number of decompositions levels are pre-defined during multi-resolution analysis.

[2]The public and private keys correspond to the popular asymmetric encryption technique [19].

[3]Resolution is used as a measure of the amount of detailed information contained in the image.

6 Partitioning Watermarking Tasks Across the Network

Let us consider how watermarking tasks can be partitioned between the proxy server and the client device. There is a trade-off between the degree of security available on the mobile devices versus its energy consumption. In the next two sections, we consider two scenarios that we have experimented with. Note that, in the context of our system architecture (shown earlier in Figure 1), the mobile devices may extract watermarks from received image data or may embed watermarks in images that they send out.

6.1 Mapping all Watermarking Tasks to the Proxy

In an attempt to maximize battery life, we "offload" all the image and watermark preprocessing and embedding tasks to the proxy. The only extra communication required is transmitting the watermark and the secret key once from the handheld to the proxy. Whether the handheld is transmitting or receiving, the data is streamed via the proxy anyway, so the proxy can do watermark embedding and extraction and authentication.

This watermark process migration is applicable in office environments where a *trusted* proxy can act as an "agent" or representative for the mobile device and can take care of authentication and quality of service negotiation with the content server. In such a scenario the handheld and the proxy may be connected using a secure SSL connection that prevents watermark interception and tampering.

6.2 Security-driven Partitioning

A more secure partitioning scheme for both watermark embedding and extraction requires some participation from the device in the watermarking process.

During watermark embedding, we migrate the following tasks to the proxy: bit decomposition, coefficient calculation using DWT, and watermark coefficient encryption using the public key. So, the handheld first sends the image and the watermark to the proxy. The proxy processes them

and sends the image and watermark coefficients back to the handheld. The handheld then embeds the watermark coefficients into the image using a unique coefficient relationship to generate the watermarked image.

During watermark extraction, the handheld extracts the watermark coefficients from the watermarked (received) image and use its private secure key to decrypt the watermark coefficients. These coefficients are then sent to the proxy to do inverse DWT et cetera to generate the original watermark. The proxy then sends this watermark back to the handheld for authentication against the original watermark. We discuss the communication overhead of this partitioning scheme in the results (see Section 7.3).

This is a secure approach since during watermark embedding, the proxy does not know the coefficient relationship used to embed the watermark coefficients in the image. Similarly during watermark extraction, the proxy does not have access to the private secret key.

Arguably, in this partitioning scheme, the original watermark (that is given to the proxy) is open to a malicious attack by the proxy. One way to guard against this is to scale or multiply the extracted coefficients by a prime number. Thus, only the handheld knows that the resultant watermark (actually embedded) is different from the original one. Consequently, watermark extraction will lead to a scaled watermark that the handheld can downscale and compare with the original watermark it has. If the proxy tries to return the original (unscaled) watermark during watermark extraction, the handheld will immediately detect the tampering, since the proxy does not know the multiplication factor.

7 Experimental Setup and Results

In this section, we present experimental results for energy consumption and performance for the various approaches discussed in Section 6.

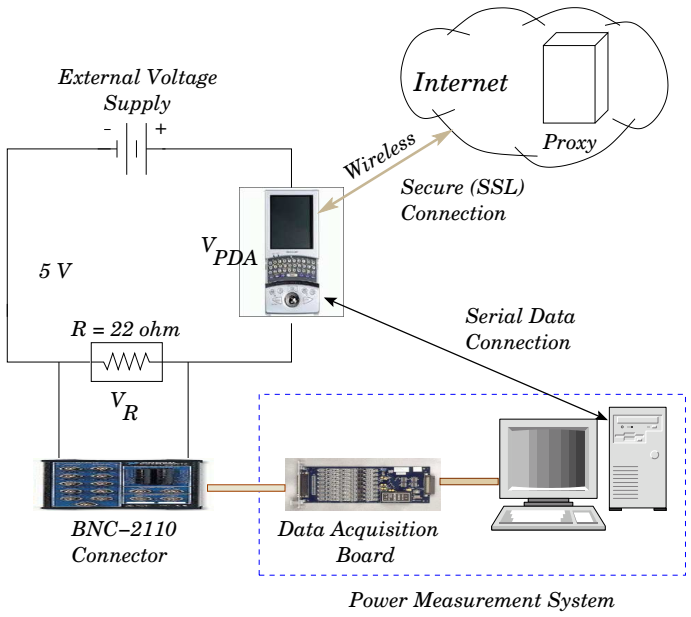

Figure 4: Experimental Setup

7.1 Setup

Our experimental setup is shown in Figure 4. 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 compact flash network card on the Zaurus for communication. We removed the internal battery from the Zaurus and placed a resistor in series with the power supply, as shown in Figure 4. 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 :

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

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

Algorithm	Description
Corvi [21]	Embeds the scaled watermark bits into the image
Dugad [22]	Embeds a watermark of n normally distributed values into the co-efficients of a sub-band
Kim [23]	Embeds a sequence of watermarks into sub-bands in each decomposition level
Wang [24]	Embeds a watermark into selected sub-bands determined by a threshold
Xia [25]	Embeds a watermark into the wavelet transformed image along different directions
Xie [26]	Employs a 3-pixel window to embed a watermark
Zhu [27]	Embeds a watermark into a sub-band of a decomposition containing the largest coefficient

Table 1: Characteristics of the Watermarking Algorithms

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 ($T = 1/1000$). 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.

To evaluate the effectiveness of our partitioning scheme, we tried it across a number of different algorithms for image watermarking. Specifically. we performed our experiments on 7 wavelet-based image watermarking algorithms discussed in [28] available from [29]. The source code was cross-compiled using *arm-gcc* for Zaurus. In Table 1 we list a brief description of each of these watermarking algorithms. We also measured the execution time of the watermarking algorithms

Algo.	Embedding			Extraction		
	Exec. time(s)		Energy	Exec. time(s)		Energy
	Proxy	PDA	Gain (J)	Proxy	PDA	Gain (J)
Corvi	0.79	211.58	129.29	0.74	213.96	99.76
Dugad	0.75	212.50	106.84	0.34	98.10	47.68
Kim	0.83	217.73	113.65	0.79	200.92	110.13
Wang	0.84	217.13	132.88	0.74	198.70	118.23
Xia	0.85	202.08	135.90	0.78	191.46	109.57
Xie	0.77	211.87	223.01	0.38	97.02	92.26
Zhu	0.82	222.59	252.88	0.71	204.45	236.33

Table 2: Execution time and energy analysis when all the watermarking tasks are offloaded to the proxy versus when they are run on the PDA.

on a proxy server, an Intel® Celeron® 1.70 GHz Linux PC machine, connected to the PDA over a 802.11 wireless network.

7.2 Mapping all Watermarking Tasks to Proxy

Table 2 lists the execution time for watermark embedding and extraction for the algorithms listed in Table 1. Columns 2 and 3 correspond to the execution time of the watermark embedding process (for a gray-level image of size 512×512) on the proxy and the PDA respectively and columns 5 and 6 for the watermark extraction process. Columns 4 and 7 list the energy gain or savings achieved by "offloading" watermark embedding and extraction to the proxy. This is in effect the energy required to execute the entire watermarking embedding and extraction algorithms on the handheld. We fine that energy savings vary from $47J$ to a high of $236J$.

We also note from the execution time results that migrating the entire watermarking procedure

Algorithm	Embedding Exec. Time(s)			Extraction Exec. Time(s)		
	Proxy	PDA	Total	Proxy	PDA	Total
Corvi	0.69	0.36	1.05	0.63	0.41	1.04
Dugad	0.68	0.19	0.87	0.31	0.15	0.46
Kim	0.70	0.53	1.23	0.65	0.48	1.13
Wang	0.69	0.56	1.25	0.67	0.64	1.31
Xia	0.69	0.46	1.15	0.61	0.55	1.16
Xie	0.68	0.41	1.09	0.31	0.18	0.49
Zhu	0.70	0.65	1.35	0.64	0.64	1.28

Table 3: Execution times when the watermarking tasks are partitioned between the proxy and the handheld without compromising security.

to the proxy also boosts *performance* by over two order of magnitude. Thus, the dual performance and energy benefit makes this partitioning highly lucrative for environments where the handheld communicates with a trusted proxy over a secure channel such as SSL.

7.3 Security-Driven Partitioning

We partitioned the watermarking algorithms as discussed in Section 6.2 and executed them using our proxy server along with the Zaurus PDA. The execution time results for the tasks run on the proxy and the PDA for watermark embedding and extraction are listed in Table 3.

The results in this table demonstrate that our security-driven task partitioning leads to very low execution times as compared to executing the watermarking on the PDA (as listed earlier in Table 2). In fact the total execution time is comparable to executing watermarking exclusively on the proxy (compare with Table 2).

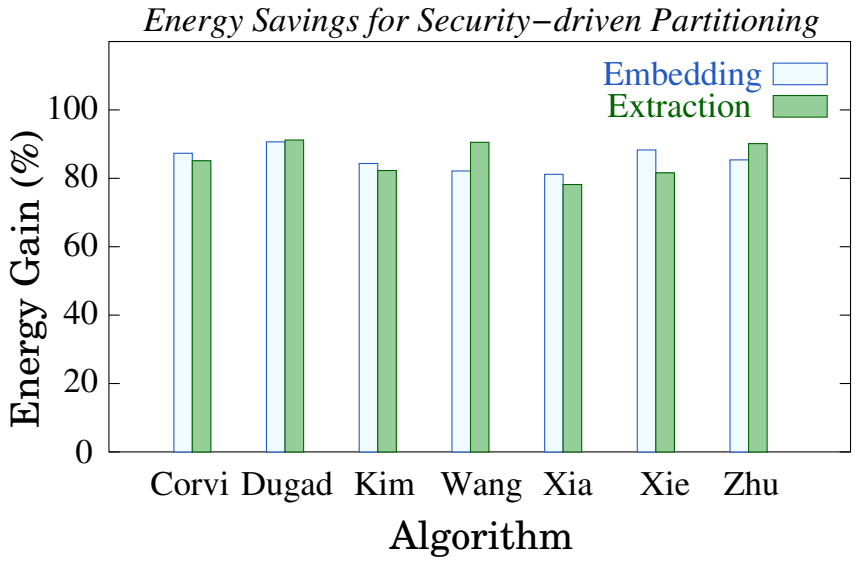

Figure 5: Energy Savings for the security-driven partitioning scheme over executing the entire watermarking algorithm on the PDA.

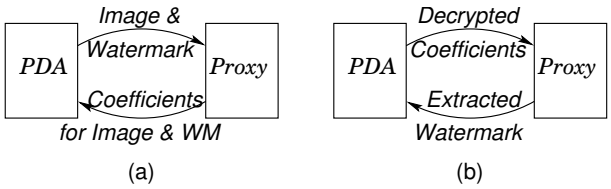

Figure 6: Communication overheads of (a) watermark embedding, (b) watermark extraction.

We present the energy savings from the security-driven partitioning over executing watermarking on the handheld in Figure 5. Clearly, we achieve large savings, over 80 %, for all the watermarking algorithms using our partitioning scheme. Note that in absolute numbers the watermarking tasks run on the PDA consume about 30 to 50 Joules.

Algo	Embedding		Extraction	
	Image Transf. Time (sec)	Assoc. Energy	Watermark Transf. Time (sec)	Assoc. Energy
Corvi	0.73	400mJ	0.11	55mJ
Dugad	0.70	393mJ	0.12	57mJ
Kim	0.74	405mJ	0.10	51mJ
Wang	0.75	410mJ	0.12	58mJ
Xia	0.69	387mJ	0.12	58mJ
Xie	0.76	415mJ	0.10	52mJ
Zhu	0.73	401mJ	0.10	50mJ

Table 4: Time and Energy overheads due to communication between proxy and PDA in the security-driven partitioning scheme.

7.4 Communication Overhead

The communication overheads due to the security-driven partitioning scheme are shown diagrammatically in Figure 6. During embedding the PDA sends the image and watermark to the proxy for pre-processing. The proxy sends the coefficients back to the PDA. During extraction, the PDA sends the decrypted coefficients to the proxy, who then sends back the extracted watermark. Among these, the only significant communication overheads are due to the transfer of the image and watermark between the PDA and the proxy. We list the transfer times and communication energy associated with these transfers during watermark embedding and extraction in Table 4. These overheads are very small compared to the watermarking tasks that run on the PDA (0.4 Joules for transfer versus about 30-50 Joules for the watermarking process). The coefficients of the image and the watermark are smaller than half the size of the image and watermark; hence, they add only a small energy overhead to the partitioned process.

8 Conclusion

We analyzed wavelet-based image watermarking techniques and proposed task partitioning schemes of the watermarking tasks between a proxy server and a handheld device. One way to minimize the energy usage of watermarking on the handheld is to move all the watermarking tasks to the proxy – this is applicable only in a very secure and trusted environment. We proposed a more secure approach in which we partitioned the tasks such that energy consumption on the handheld is minimized without compromising the security of the watermarking process. We demonstrated through experimental results and energy measurements of several wavelet-based image watermarking algorithms on a PDA that our security-driven partitioning scheme leads to an average 80% reduction in the energy, while achieving over two orders of magnitude improvement in performance over executing watermarking on the handheld. Clearly, no approach that relies on network communication and/or another computer for offloading computation can be 100% secure. However, we believe that our partitioning scheme is a balance between reducing energy and securing the data. In future work, we plan to study an energy and security-aware partitioning scheme for video watermarking.

9 Acknowledgements

This work is supported in part by NSF Grant ACI-0204028. We would like thank R. Cornea, Paolo D'Alberto for reviewing the paper.

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] F. P. 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] http://www.digimarc.com.

[6] Jim Bauerle et. al. Multimedia applications : Imaging, video, and security.

[7] Verance. *http://www.verance.com/*.

[8] http://www.real.com.

[9] D. V. Thanh. Security issues in mobile ecommerce. In *Proc. of Int'l Workshop on Database and Expert Systems Applications*, pages 412–425, Sep 2000.

[10] 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.

[11] 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.

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

[13] 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.

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

[15] 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.

[16] 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.

[17] H. Inoue, A. Miyazaki, A. Yamamoto, and T. Katsura. A digital watermark based on the wavelet transform and its robustness on image compression. In *IEEE International Conference in Image Processing*, pages 391–395, Oct 1998.

[18] F. Hartung, J. K. Su, and B. Girod. Spread spectrum watermarking: Malicious attacks and counterattacks. In *Security and Watermarking of Multimedia Contents*, pages 147–158, Jan 1999.

[19] A. Gordon and A. Jeffrey. Types and effects for asymmetric cryptographic protocols. In *Proc. CSFW*, 2002.

[20] I. Daubechies. Orthonormal bases of compactly supported wavelets. In *Comm. Pure & Appl. Math.*, pages 909–996, 1998.

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

[22] 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.

[23] J. R. Kim and Y. S. Moon. A robust wavelet-based digital watermark using level-adaptive thresholding. In *Proceedings of the 6th IEEE International Conference on Image Processing*, Kobe, Japan, Oct 1999.

[24] H. M. Wang, P. Su, and C. J. Kuo. Wavelet-based digital image watermarking. In *Opt. Express*, pages 491–496, Dec 1998.

[25] X. G. Xia, C. G. Boncelet, and G. R. Arce. Wavelet transform based watermarking for digital images. In *Opt. Express*, pages 497–511, Dec 1998.

[26] L. Xie and G. Arce. Joint wavelet compression and authentication watermarking. In *Proceedings of the 5th IEEE International Conference on Image Processing*, Chicago, IL, Oct 1998.

[27] W. Zhu, Z. Xiong, and Y. Q. Zhang. Multiresolution watermarking for images and video: A unified approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):545–550, 1999.

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

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