Securing Multimedia Content Using Joint Compression and Encryption

Amit Pande and Prasant Mohapatra
University of California, Davis

Joseph Zambreno
Iowa State University

A joint multimedia compression and encryption technique can significantly reduce the computational requirements of video processing systems while also preserving the properties of compressed video.

Algorithmic parameterization and hardware architectures can ensure secure transmission of multimedia data in resource-constrained environments such as wireless video surveillance networks, telemedicine frameworks for distant healthcare support in rural areas, and Internet video streaming. Joint multimedia compression and encryption techniques can significantly reduce the computational requirements of video processing systems.

Wireless video surveillance networks have been recently deployed in different network settings such as WiFi, WiMAX, and wireless sensor networks. These networks, which are deployed in private and public settings, can carry sensitive visual information. For example, the live feeds from more than 10,000 cameras are used by city police departments in New York and Chicago to monitor criminal activity. It is important to protect the video feeds of these cameras from eavesdropping. However, providing real-time end-to-end encryption of video data using conventional cryptographic primitives is difficult because of wireless network characteristics (low bandwidth, frequent packet drops), quality of service concerns (real-time delivery, low jitter), and the limited computational resources at the encoder. Apart from the need for a secure way to transmit the videos, we need computationally efficient algorithms to save computing power and enable easy access of visual information from encrypted videos in databases.

Telemedicine is an application of clinical medicine where consultation as well as remote medical procedures and examinations are performed using interactive audio-visual media. Extending such services to remote locations that lack high-speed connections and even electric power in underdeveloped countries requires efficient low-power devices. Furthermore, the privacy of patient information and prescriptions is an important concern for these applications, considering the vulnerability of communication channels to eavesdropping and other attacks.

The advent of embedded multimedia systems has already revolutionized the way we live. Video messaging, videoconferencing, video surveillance, and Internet video sites such as YouTube are becoming increasingly popular and pervasive. Network traffic in next-generation cellular networks is predicted to be dominated by video,[1] and it makes sense to provision for video security in these applications. Most mobile devices have low computational resources and limited battery resources.

Conventional encryption schemes, however, such as those using the Advanced Encryption Standard (AES) and Data Encryption Standard (DES) are not suitable for video data because of the large computational overhead. Compressed multimedia streams exhibit well-defined hierarchical structure that can be exploited in several useful ways (such as for scalability, random access, transcoding, and rate shaping) in low- and variable-bandwidth scenarios, but these structures are not recognizable in traditional cipher text.

To address these limitations, we discuss the design of algorithms and hardware architectures for secure transmission of multimedia data in resource-constrained environments. Specifically, an augmented video coding model for joint compression and encryption can be used that can significantly reduce the computational requirements. In this work, we propose

1070-986X/13/$31.00 © 2013 IEEE

Published by the IEEE Computer Society

building design blocks that enable security for these applications at the algorithmic level and leave domain-specific optimization to application developers. These algorithmic optimizations map easily to fixed-point hardware, allowing us to create efficient architectural optimizations for resource constrained scenarios. In other application scenarios, these approaches can complement the security provided by conventional schemes such as AES.

The proposed schemes are low cost in the sense that the required computational hardware is considerably smaller than existing approaches and, in some configurations, require fewer hardware resources than conventional video compression schemes.

Basics in Multimedia Compression

Multimedia compression involves large computations and a large number of data transfers, thus requiring application-specific hardware such as ASICs and FPGAs to compress and deliver the media in real time. Video compression using hardware accelerators has increasingly gained attention because of the popularity of low-power embedded devices. Thus, an efficient architectural design of multimedia compression blocks is a must to ensure real-time video delivery.

Although compressed multimedia files typically exhibit a well-defined hierarchical structure that can be exploited in several useful ways (such as for scalability, random access, transcoding, and rate shaping), these structures are not recognizable in cipher text and, hence, are wasted. These properties are useful to index, search, and retrieve compressed multimedia from digital libraries and also for communication over heterogeneous networks. We need a paradigm in which encryption does not change the compressed output but provides access and copy control for concerned media. Thus, we need encryption of video data that does not affect the properties of the compressed bitstream or the compression performance. On the one hand, compression and encryption operations require a large amount of computational overhead even as we have seen an increasing trend toward battery-driven low-power embedded systems such as portable mobile devices (iPods, mobile phones, and cameras).

Therefore, in addition to optimizations in hardware architectures, we also need to reduce the computational cost of secure multimedia transactions through algorithmic improvements.

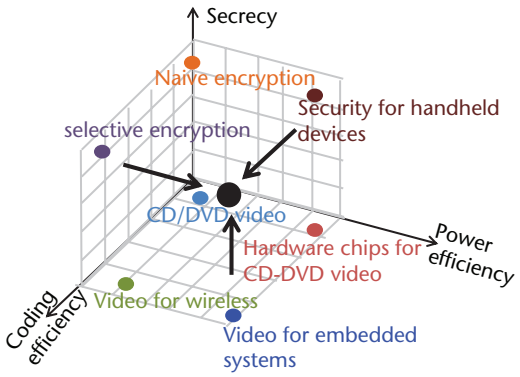

Figure 1. The broad goal of a joint approach is to develop algorithms and architectures to push the operating curve toward (1, 1, 1), considering all the factors (coding efficiency, power efficiency, and privacy) together during design.

Figure 1 illustrates the motivation behind our proposed approach: considering coding efficiency, power efficiency, and privacy in a joint design of algorithms and architectures. (See the "Related Research in Multimedia Compression" sidebar for more details on previous, related efforts.)

Our Approach

In this work, we propose a redesign of the video compression blocks themselves to enable encryption and efficient mapping onto hardware. For example, if the video coders have an additional parameter that can be changed to provide encryption, we can use it as a key space for secret key generation. The required mixing of the inputs, as required by cryptographic ciphers, is automatically provided by different blocks of the video coding system. Similarly, if we could design the system with the rational coefficients as a design constraint, we will obtain a hardware-amenable implementation. The redesign of video coding blocks enables joint compression and encryption and reduces the computational requirements of multimedia encryption algorithms. The approach modifies the compression system properties instead of the compressed bitstream itself. Moreover, the redesign is amenable to hardware acceleration over reconfigurable computing platforms. We leverage signal processing techniques to make the algorithms suitable for hardware optimizations (and encryption) and reduce the critical path of circuits using hardware-specific optimizations.

A trivial way to explain this solution (of joint encryption and compression) is to find 2^N different but similar ways to compress a video, where all of them give similar compression performance and the compressed bitstream has the same properties. For large values of 2^N, we can say that the N-bit code representing the

Related Research in Multimedia Compression

For the last five decades, the research in video coding has been commercially utilized in the form of state-of-the-art video coding standards such as MPEG-1 and MPEG-2. MPEG-2 based schemes are useful for DVD quality compression. In these scenarios, coding or power efficiency are not constraints, and security is provided using end-to-end encryption with AES or some variant.[1] Some work has been done for resource optimization in these schemes in cases of low bandwidth, but the problem is not as acute as with cases involving battery-driven low-power embedded systems.

Recent research in wireless networks and video surveillance aims at optimizing video quality for wireless transmission and often uses the MPEG-4 format,[2,3] which produces a more compressed and scalable bitstream. The H.264 scalable video coding (SVC) format is the most recently used in this work. Many researchers have also tried to optimize the hardware implementations of MPEG-2 and MPEG-4 based video applications.

Recent research in video encryption over wireless and other scarce resource channels has identified the need for nontraditional approaches to video encryption besides the use of standard cryptographic ciphers. These approaches involve selective or partial encryption of the video stream, chaotic encryption, and shuffling in compressed bitstreams. There has also been research on accelerating these video processing kernels in hardware such as ASIC or FPGA.

Thus, little research has targeted the three-fold goal of high compression, low computational cost, and secrecy. With these three goals in mind, we propose our approach.

References

1. M. Baugher et al., *The Secure Real-Time Transport Protocol (SRTP)*, RFC 3711, The Internet Soc., Mar. 2004.
2. G. Gualdi, A. Prati, and R. Cucchiara, "Video Streaming for Mobile Video Surveillance," *IEEE Trans. Multimedia,* vol. 10, no. 6, 2008, pp. 1142–1154.
3. D. Hu et al., "Scalable Video Multicast in Cognitive Radio Networks," *IEEE J. Selected Areas in Comm.,* vol. 28, no. 3, 2010, pp. 334–344.

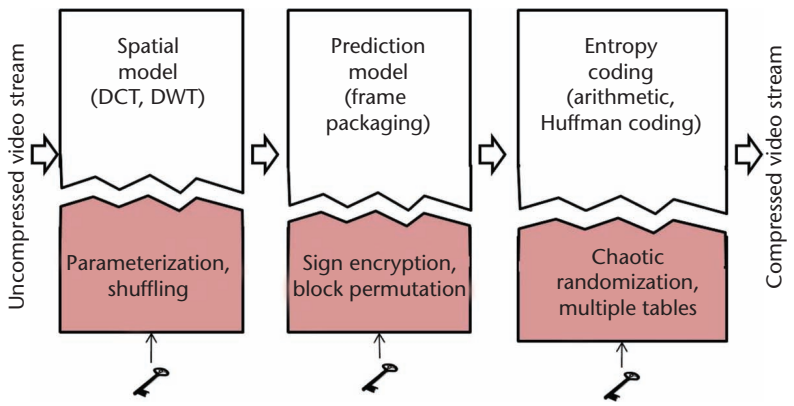

Figure 2. Video compression system augmented with different operations to ensure real-time encryption.

choice of compression system is the encryption key of the system. For such a system to be secure, the combined system must follow cryptographic requirements such as good diffusion and confusion properties.[2] The output from two closely related keys should be nearly uncorrelated, and there should not be a way to reverse engineer the N-bit key except by a brute force attack.

This proposal also meets the requirements of property-preserving encryption because essentially we are trying to shuffle the compression parameters using the key and not modifying the input bitstream itself. Each of the 2^N compression systems provide "property-preserving" compression.

We can augment the encryption in the video coding system by redesigning individual video coding blocks followed by integration into a single prototype and hardware implementation (see Figure 2). Here is a brief description of these modifications:

- *Augmented prediction model*: We propose using a fuzzy prediction model, which selects from several past and future frames and uses multiple streams, based on a key-dependent fuzzy logic instead of the traditional use of immediate neighbors. Similarly, the sign bits of the motion vectors can be encoded and/or a key-based nonlinear mapping of motion vectors can be performed.
- *Augmented spatial model*: We propose parameterizing the transform filter (discrete wavelet transform, DWT) so that the choice of filter depends on the key value. Different filters give different output coefficients while the compression efficiency of each is similar. The output subbands of DWT (or subblocks of DCT) can be reoriented and permuted according to a key.
- *Augmented entropy coding*: Modified entropy coders can be used with multiple statistical models so that the exact model choice is governed by a key. Similarly, arithmetic coding can be implemented using a key-based chaotic random map. The reiterations of the

Table 1. Hardware-amenable implementation of DWT.

Features	Daubechies 9/7	Poly-DWT*	Tay, 01	Kotteri, 05	Chao, 03	Martina, 07	Martina, 05
Adders	15	**9**	19	15	8	19	21
Multipliers	9	**0**	0	0	4	0	0
Clock(MHz)	107	**389**	–	–	–	200	–

* Poly-DWT yielded a more than 50 percent improvement in hardware requirements with increased clock frequency (using a Xilinx Virtex-V XC5VLX30 FPGA testbed).

chaotic map make the output appear random, while the choice of the map itself is governed by a key.

- *All-on-a-chip*: Hardware-specific optimizations on augmented modules will enable us to fit the prototype onto a single chip.

Compression Module Details

In this section, we discuss two augmented compression modules to make them amenable to encryption and hardware implementation. The augmented frequency transform is illustrated with the help of a DWT, which is used in a MotionJPEG2000 coder, while augmented entropy coding is shown using arithmetic coding (AC) that is used in H.264 and other formats.

Augmented Frequency Transform

The efficient representation of time-frequency information by the DWT has led to its popularity for spatial modeling. DWT provides superior rate distortion and subjective image quality performance over existing standards. Many image and video compression schemes have been derived from DWT-based structures that have become increasingly popular because of its excellent compression properties. The 1-D DWT can be viewed as a signal decomposition using specific low-pass (H_0) and high-pass (H_1) filters. A single stage of image decomposition can be implemented by successive horizontal row and vertical column wavelet transforms. To recover the image, we perform the inverse DWT using another set of low- and high-pass filters G_0 and G_1, respectively. The two most common DWT filters are Le Gall's 5/3 filter and Daubechies 9/7 filter.[3]

Existing Hardware-Amenable Implementation Efforts. Rational binary coefficients for DWT have been designed to help achieve a multiplier-free implementation of DWT filter coefficients. However, these multiplier-free implementations involve image reconstruction quality trade-offs. Many other researchers have also faced the problem of reducing DWT complexity.

Augmenting the DWT for Hardware Implementation. Rather than optimizing the mapping of filter coefficients, we want to rederive the filter coefficients that map better to custom hardware.

For example, the quadrature mirror filter (QMF) properties of DWT allow perfect reconstruction of an image after inverse DWT operation at the decoder. The condition for perfect reconstruction of an image, when applying DWT, boils down to $G_0(z)\,H_0(z) + G_0(-z)\,H_0(-z) = 2$.

Solving this equation using Lagrange half-band filters (LHBF) leads to the Daubechies filter, which is the most widely used DWT filter. However, these coefficients are irrational and lead to inefficient hardware implementations.

David Tay derived rational coefficients for DWT, setting this as a constraint while solving these equations.[4] In our preliminary work, we used this result to build a polymorphic DWT (Poly-DWT) architecture that uses binary rational coefficients, which are amenable to hardware implementation. We also added some features that help Poly-DWT provide a dynamic response to changing external conditions and thus dynamically adjust video quality and power requirements.

The hardware prototype of the proposed system on a Xilinx Virtex-V XC5VLX30 FPGA has the following features for a hardware-amenable implementation:[5]

- The new architecture enables dynamic allocation of hardware resources to efficiently create a dynamic response to changing external conditions.
- Our architecture is multiplier-free. Furthermore, its hardware requirements (nine adders) are nearly 50 percent that of existing architectures in the research literature (see Table 1),

*Table 2. Improved image reconstruction (PSNR values) with hardware-amenable DWT implementation.**

	Bitrate = 0.5 bpp			Bitrate = 2 bpp		
Image	Daubechies 9/7	Poly-DWT	Martina, 07	Daubechies 9/7	Poly-DWT	Martina, 07
Lena	28.21	**29.46**	27.70	38.47	**38.17**	36.50
Surveillance	26.10	**28.10**	26.54	38.41	**42.21**	39.21
Lecture	34.35	**33.80**	32.73	48.30	**51.25**	43.71
Helicopter	33.75	**35.70**	35.01	48.59	**54.72**	47.14

* Peak signal-to-noise ratio (PSNR) values.

while its image compression performance is better than a state-of-the-art fixed-point implementation (see Table 2).
- A scheme to allow runtime switching between 5/3 and 9/7 wavelet structures was proposed. Our architecture enables on-the-fly switching of hardware resources to suit the power budget of the video processing system.

Augmenting the DWT for Both Encryption and Hardware. Dominik Engel and Andreas Uhl parameterized DWT,[6] but the resulting key space is small and restricted. Moreover, they do not discuss any hardware implementation issues. Motivated by the successful parameterization of DWT for hardware implementation, we investigated the use of parameterization for encryption purposes.

The previous parameterization has only two to three rational points, but for use in parameterization for encryption, we need a family of rational coefficient filters. Zaide Liu and Nanning Zheng presented a parameterized construction of the filters typically used for image compression.[7]

We obtain a new parameterization with two interesting features. First, it has a free parameter α that can be varied as a key parameter without sacrificing the perfect reconstruction property of the video stream. The terms of α are nonlinear, which makes it difficult to recover the α value from the transform output. Second, all other coefficients are rational and amenable to hardware implementation. These properties, along with the property of the subband rotation, were used to design a video encryption scheme optimized over hardware referred to as the secure wavelet transform (SWT).[8] A key space of $25N + 3$ bits can be obtained from N levels of wavelet decomposition. For an image size of 512×512 pixels, the upper limit of N is 9.

The hardware prototype of the proposed system on a Xilinx Virtex-V XC5VLX330 FPGA has the following features for hardware-amenable implementation:

- The DWT kernel was parameterized to incorporate the encryption feature and promise reasonable security for real-time embedded multimedia systems.
- A zero computation overhead subband reorientation scheme added to parameterization leads to efficient image encryption (see Figure 3).
- An optimized hardware implementation of the SWT architecture is presented. The proposed hardware implementation has a low critical path and thus achieves a high clock frequency (see Table 3).

Augmented Entropy Coding

Entropy coding schemes are used to compress data, in a lossless manner, to a maximum level with the assumption of an independent and identically distributed random variable distribution. The two most popular entropy coding techniques are Huffman coding and arithmetic coding (AC). Of these, Huffman coding is computationally cheap, while AC yields better compression. AC involves recursive partitioning of the range [0,1) in accordance with the relative probabilities of the occurrence of the input symbols.

Existing Efforts in AC-Based Encryption. An earlier work proposed a chaos-based adaptive arithmetic coding technique.[9] The arithmetic coder's statistical model is made varying in nature according to a pseudo-random bitstream generated by coupled chaotic systems. Many other techniques based on varying the statistical model of entropy coders have also been proposed in the literature, but these techniques

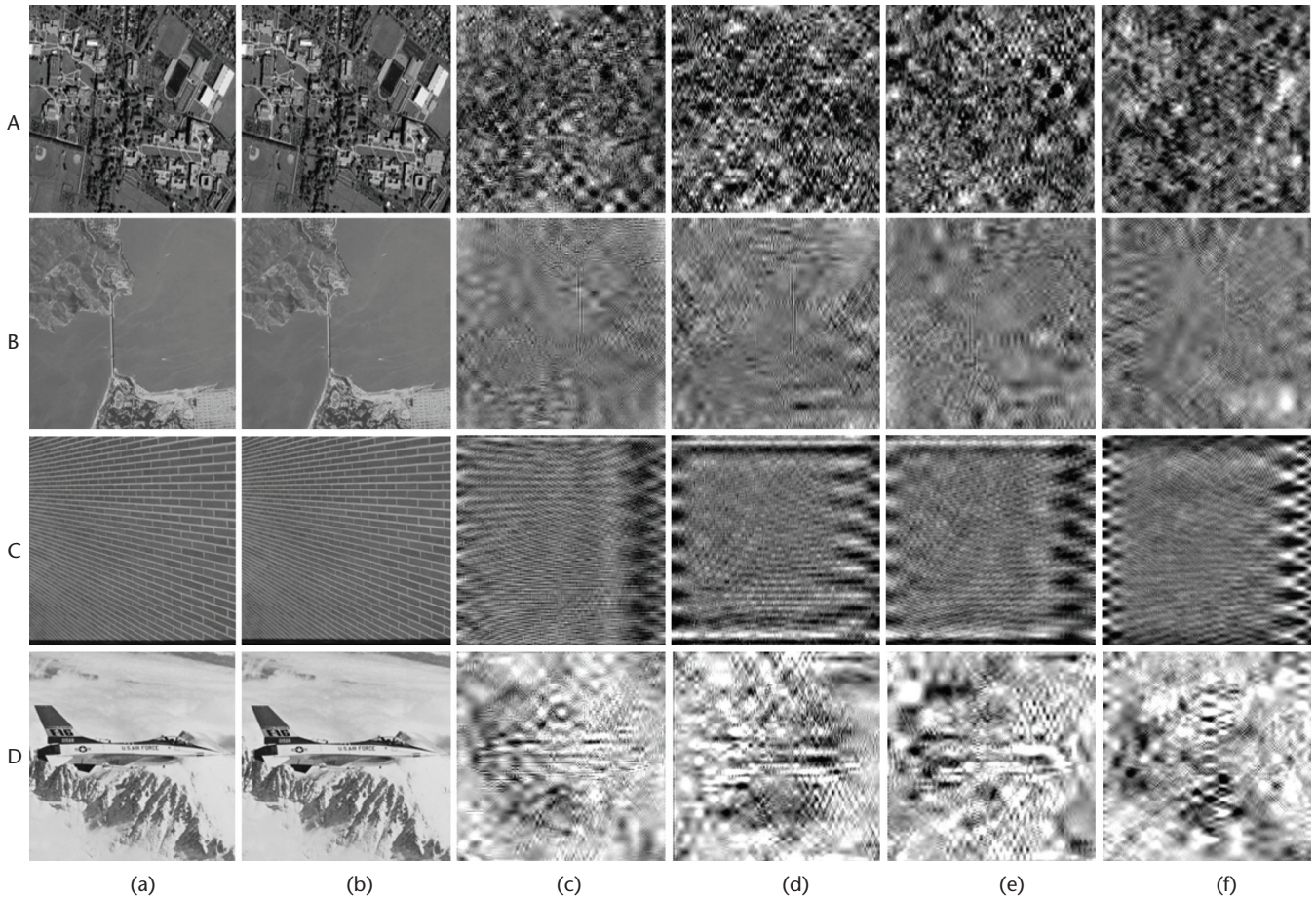

Figure 3. Video encryption with augmented DWT (a) Original image encrypted with key-0, (b) image decrypted with same key, and (c)–(f) image decrypted with randomly generated keys.

*Table 3. A first architecture for video encryption on DWT.**

Features	SWT	Martina, 07	Daubechies 9/7	Jou, 01	Vishwanath, 95	Huang, 04
Multiplier	0	16	13	36	36	12
Adder	11	19	15	16	36	16
Critical path	$4T_a + T_l$	$5T_a$	$T_m + 4T_a$	$T_m + 2T_a$	$T_m + 4T_a$	$4T_m + 8T_a$
Frequency	114	200	107	–	–	–
Encryption	Yes	No	No	No	No	No

* A Xilinx Virtex XCVLX330 FPGA was used as the testbed. T_m, T_l, and T_a are the time delay in multiplier, look-up table, and adder circuits, respectively.

suffer from losses in compression efficiency that result from changes in entropy model statistics and are weak against known attacks.[10]

Recently, Marco Grangetto and his colleagues presented a randomized arithmetic coding (RAC) scheme that encrypts by inserting some randomization in the arithmetic coding procedure at no expense in terms of coding efficiency.[11] RAC needs a key with a length of 1-bit per encoded symbol. Hyungjin Kim and his colleagues presented a generalization of this procedure called secure arithmetic coding (SAC).[12] The SAC coder builds over a key-splitting arithmetic coding, where a key is used to split the intervals of an arithmetic coder, adding input and output permutation to increase the coder's security. Successful attacks have been demonstrated against these SAC schemes.

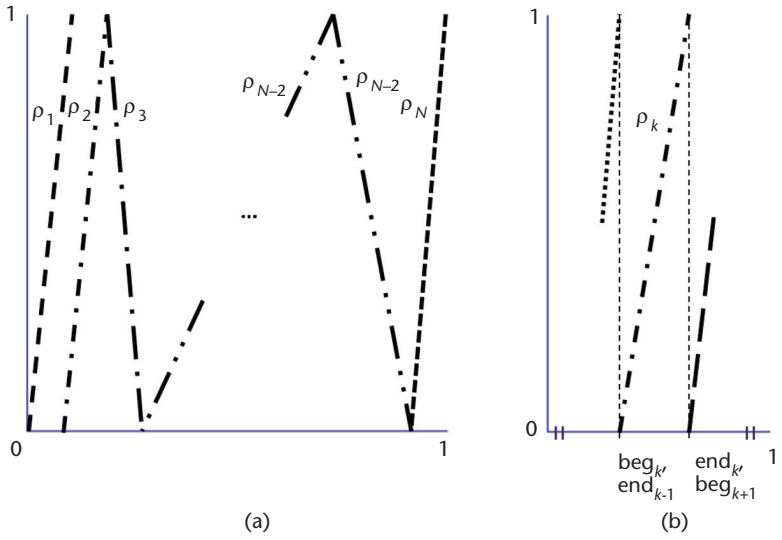

Figure 4. A sample piecewise linear map for arithmetic coding such as compression. (a) The entire map (ρ) and (b) a single linear part of the map (δ_k). It can have a positive or negative slope.

Augmenting Encryption to AC. We generalize the arithmetic coder (as we did in the case of DWT) to get multiple ways of encoding without losing compression efficiency. Nithin Nagaraj and Prabhakar G. Vaidya noted the equivalence between AC and chaotic maps.[13] We first interpreted AC in terms of iterations over piecewise linear chaotic maps and then defined a family of such maps, each yielding the same compression efficiency. We next developed a data (image or video) encryption scheme based on AC that we call chaotic arithmetic coding (CAC). The CAC scheme uses a key to make the exact choice of map from the family of predefined maps to perform AC.

Consider a scenario where we have a string $S = x_1, x_2, \ldots x_N$ consisting of N symbols to be encoded. The probability of occurrence of a symbol s_i, $i \in 1, 2, \ldots n$ is given by p_i such that $p_i = N_i/N$ and N_i is the number of times the symbol s_i appears in the given string S. We next consider a piecewise linear map (ρ) that is defined on the interval [0, 1) to [0, 1). It can be decomposed into N piecewise linear parts such that each part maps the region on the x axis [beg_k, end_k) to the interval [0, 1). This mapping is one-one and onto. Each linear map is associated uniquely with one symbol and does not intersect with another. The mapping between the linear map and the assigned symbol from a dictionary is defined arbitrarily but a one-one relationship must hold. The width of the map (on the x axis) is equivalent to the probability of the assigned symbol, leading to Shannon optimal compression efficiency for large strings.

Figure 4 shows a sample map fulfilling these properties. Figure 4a shows the full map with different parts $\delta_1, \delta_2, \ldots, \delta_N$ present, and Figure 4b zooms into individual linear part δ_k. The maps are placed adjacent to each other so that each input point is mapped into an output point in the range [0, 1).

There are N different piecewise maps (for encoding a symbol from a dictionary of N symbols, called N-ary AC), each with two possible orientations (with a positive or negative slope). Thus, the number of total permutations possible is given by $N!2^N$. Thus, for N-ary arithmetic coding or arithmetic coding with N symbols, it is possible to have $N!2^N$ different mappings, each leading to the same compression efficiency. Because we can arbitrarily choose any of the $N!2^N$ maps, the key space for encoding a single bit of data is ($\lceil \log_2 N!2^N \rceil$) bits, where $\lceil\ \rceil$ represents the greatest integer function. For $N = 2$, it gives eight mappings. If we increase N to four, this value increases to 384. Thus, without any sacrifice in computational efficiency or coding rates, CAC is able to achieve a huge key space that can be effectively used for data encryption.

AC is more commonly implemented in binary mode to reduce the computational requirements of video coders. There are eight equivalent modes of skewed binary maps that can be used for binary CAC (BCAC).

Implementation Efficiency. For a normal binary arithmetic coder, at each iteration the starting interval [I_s, I_e) is updated at one end. When encoding a 0, the final interval becomes [$I_s + p(I_e - I_s)$, I_e), and when encoding a 1, the final interval becomes [I_s, $I_s + p(I_e - I_s)$). Thus, every iteration requires one multiplication and two addition operations. The decoding procedure for a binary arithmetic coder involves updating the interval [I_s, I_e) at one end, depending on whether the last decoded symbol was a 0 or 1. Thus, every iteration again requires one multiplication and two addition operations.

For a chaotic arithmetic encoder, both ends of the interval are updated at every iteration using a linear transformation $x = my + b$, thus requiring two multiplications and two additions for encoding. The decoding is simple because it involves iteration on the chaotic map according to the linear transformation $y = nx + c$ involving a multiplication and an

addition operation. There are some additional table lookups (an 8-input LUT required for BCAC to choose the exact chaotic map) involved in chaotic coding to choose the right chaotic map at every iteration that can be efficiently implemented in software or hardware.

Thus, a CAC encode requires more computations than a BAC encode, and a CAC decode requires fewer computations than a BAC decode. Our BCAC coder is similar to a BAC coder, except that the variable slope and intercept of the lines of the chaotic map are decided by the choice of a map. These values can be mapped to lookup tables and the remaining operation can be optimized similar to BAC.

Our BCAC coder is similar to a BAC coder, except that the variable slope and intercept of the lines of the chaotic map are decided by the choice of a map.

Comparison with BAC+AES

The arithmetic operations required for 1-bit encoding or decoding using BAC is two adders and one multiplier. AES-128 bits require 40 sequential transformation steps composed of simple and basic operations such as table lookups, shifts, and XORs. It needs approximately 336 bytes of memory and approximately 608 XOR operations, or roughly 3 bytes of memory and five XOR operations per bit of encoding.

A BCAC coder requires two adders and two multipliers for encoding and only one adder and one multiplier for decoding. Thus, the hardware requirements of a BCAC coder are much less than that of BAC and AES combined. The BCAC decoder is particularly simpler than an AC decoder (even without AES), which is desired for most common video applications that involve real-time decoding in mobile and embedded devices.

Compatibility with H.264

The popular H.264 codec implements content-adaptive binary arithmetic coding (CABAC) to achieve high compression efficiency. A CABAC coder has three parts: binarizer, context modeler, and BAC coder. We can directly replace the BAC coder with a BCAC coder, as proposed in an earlier section, to introduce augmented entropy coding into H.264 without introducing any coding losses. Because there is no change in dictionary values or probability value p_i of the coder, there will be no direct effect on the video coder by replacing BAC with BCAC. (A detailed analysis of the CAC scheme is available elsewhere.[14])

Key Space Security

The key space for an augmented DWT operation depends on the number of DWT decompositions and the degree of parameterization of one filter coefficients. There are $3N + 1$ different subbands obtained by an N-level wavelet decomposition of an image/frame (19 subbands for $N = 6$). Three bits are each required to describe the orientation of each subband, leading to $9N + 3$ bits. Furthermore, we divide the filter parameter in the range [1, 3] into 2^6 values. One level of wavelet decomposition involves successive filtering with row and column filters. If we have N levels of decomposition using DWT, we can choose different α values for all $2N$ filters (represented by $12N$ bits). The SWT operation allows a key space of $21N + 3$ bits. For standard and high definition image/video content, the value of N is sufficiently large ($N > 9$) to give a large key space.

The N-bit BCAC coder requires a key space of $3N$ bits. The same key may be reused for different iterations, so the key size depends on the length of the BCAC coder.

Sample Image Encryption Scheme

In an experiment, we used BCAC and SWT schemes to encrypt only the most significant DWT coefficients for an image. We considered sample images of 512×512 pixels and encoded only the coefficients corresponding to the sixth-level decomposition using DWT. Thus, the key length for SWT is only 24 bits. The key length for BCAC was chosen as 368 bits. (Each of the 16 DWT coefficients were quantized to 23 bits and then encoded; we chose 23 bits according to the dynamic range of coefficients).

Figure 5 and Table 4 present the results for this simple experiment. It can be observed that selective encryption of 0.4 percent coefficients alone can lead to considerable degradation of perceived image quality. We performed SWT

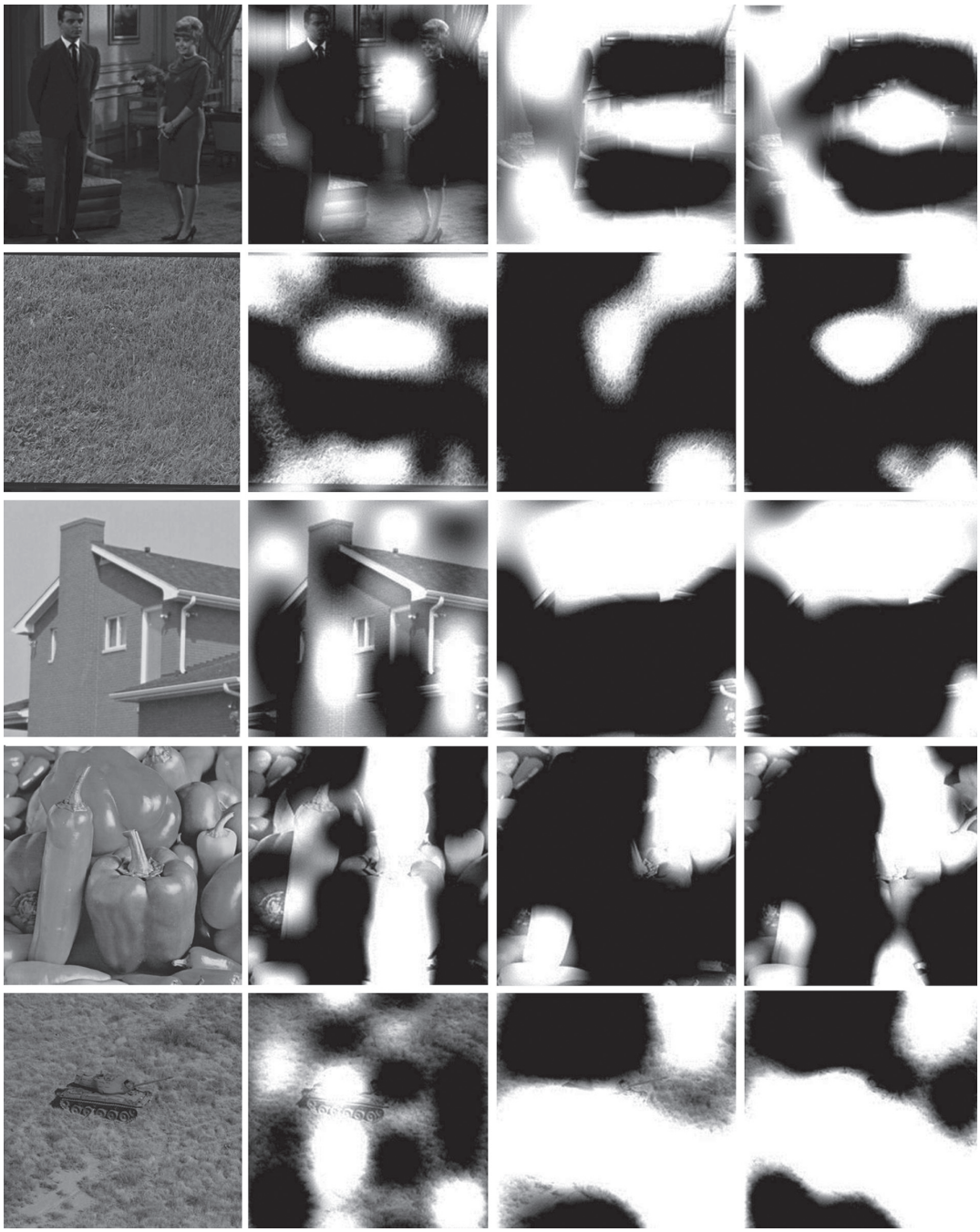

Figure 5. Image reconstruction results with selective encryption. The images were selected from the University of Southern California Signal and Image Processing Institute (SIPI) database. A selective encryption of 0.4 percent significant DWT coefficients (corresponding to sixth-level decomposition) was performed using SWT and BCAC schemes. (a) Sample images, (b) reconstruction results with SWT, (c) results with BCAC, and (d) results with SWT+BCAC.

and BCAC over the sixth-level wavelet coefficients alone (four bands of 16 × 16 pixels). The key for SWT was 24 bits, and the key for BCAC encryption was 368 bits (16 pixels values each quantized into 23 bits each, 3 bits per pixel).

The structural similarity (SSIM) metric measures the structural differences between the original image and the decrypted image. A value of 1 indicates a strong similarity to original image, while a value close to 0 indicates no similarity

*Table 4. Results for selective encryption of sixth-level DWT decomposition.**

	SWT		BCAC		SWT+BCAC	
Image	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**
Tank	0.350	4.260	0.190	−4.39	0.060	−5.33
Couple	0.147	6.819	0.070	−0.86	0.350	−1.35
Girl	0.088	2.880	0.010	−3.31	−0.020	−6.47
Grass	0.132	2.560	−0.161	−3.37	−0.132	−4.63
Peppers	0.190	0.240	0.120	8.35	0.080	5.15
House	0.160	−4.170	0.035	−4.21	0.030	−5.50

* Image reconstruction quality is evaluated for images from the University of Southern California Signal and Image Processing Institute (SIPI) database using peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) metrics.

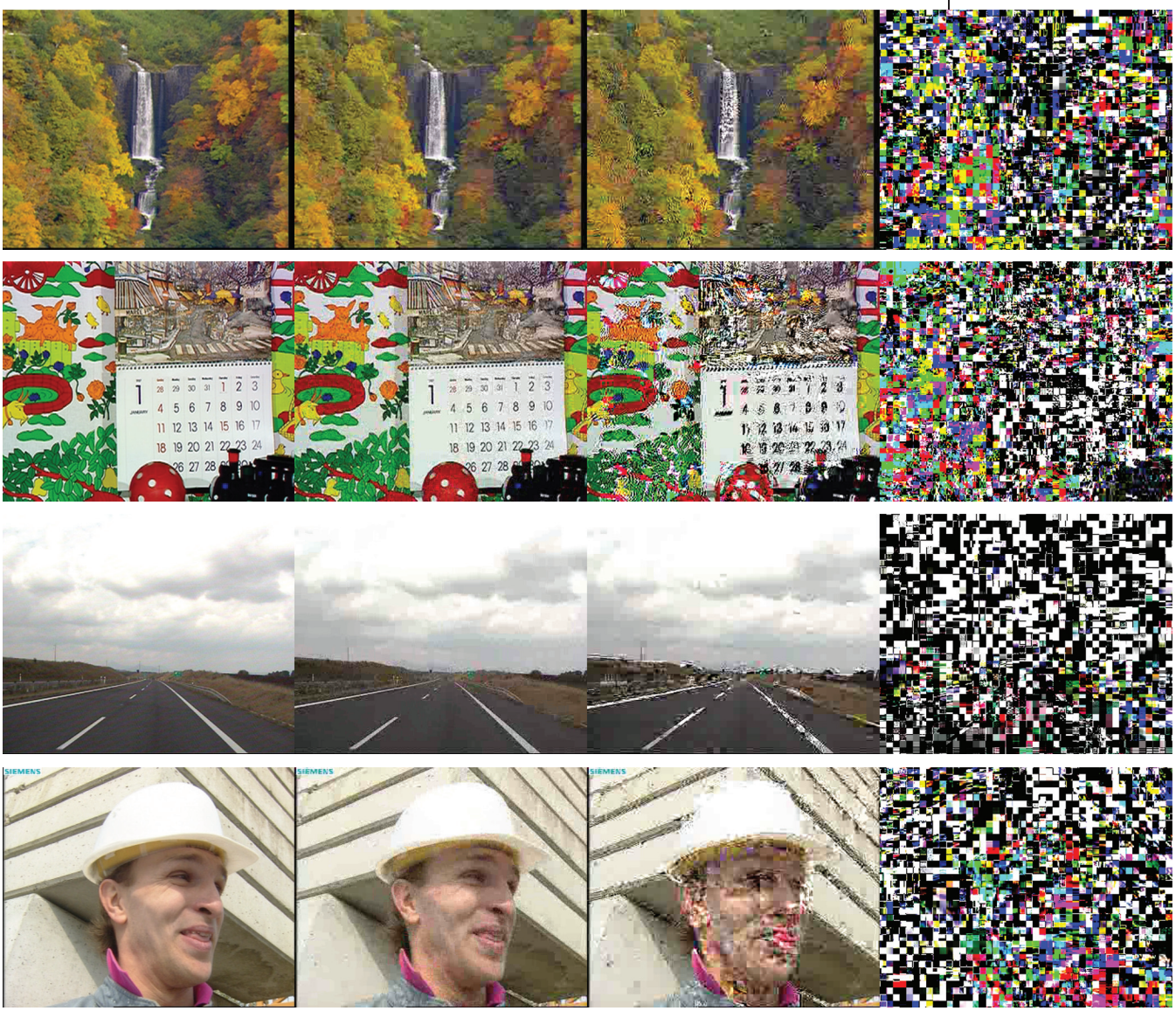

Figure 6. Video reconstruction results with the CAC scheme. The first column is the original video frames, the second column is the compressed and reconstructed video frames, the third column is only the motion vectors (MVs) encrypted with BCAC, and the fourth column is the MVs and residues encrypted with BCAC. One frame of each video sequence is shown here.

There is tremendous potential in this field for future research and deployment in state-of-the-art video codecs.

between the original and decrypted images. The peak signal-to-noise ratio (PSNR) metric measures pixel-wise differences between the two images (in decibels). A value higher than 40, for example, indicates a large similarity between the original and decrypted images, and a low value (close to 0–10 dB or less) indicates huge differences or a great deal of noise between the two images. Our results show that BCAC leads to higher degradation in the perceived image quality.

Sample Video Encryption Scheme

We used a sample video codec in Matlab that implements frequency transform (DCT) and frame prediction (*I* and *P* frames in MPEG style). We also introduced entropy coding into the system (BCAC). Figure 6 presents the results with video samples. A DCT-based block-based video codec was used with a GOP size of 10. Each video has CIF resolution. The four videos—waterfall (least motion), calendar, highway, and foreman (highest motion)—were used from standard video databases (http://trace.eas.asu.edu/yuv/). It can be seen that encrypting only the motion vectors resulted in degradation along the video regions containing motion. When we encrypt the residual also using CAC, we obtain high degrees of encryption.

Conclusion

In this work, we explored the potential of joint compression and encryption schemes for securing the multimedia content. We illustrated the potential for hardware savings and efficient encryption using this design with SWT and CAC. There is tremendous potential in this field for future research and deployment in state-of-the-art video codecs such as H.264 SVC. It is possible to develop such encryption schemes for motion compensation and estimation and implement them on embedded device architectures. **MM**

Acknowledgments

This research is supported by the National Science Foundation (under grants 2331914 and 1019343) as part of the Computing Research Association for the CIFellows Project.

References

1. O. Oyman et al., "Toward Enhanced Mobile Video Services over Wimax and LTE [Wimax/LTE Update]," *IEEE Comm. Magazine,* vol. 48, no. 8, 2010, pp. 68–76.
2. C.E. Shannon, "Communication Theory of Secrecy Systems," *Bell Systems Technical J.,* vol. 28, 1949, pp. 656–715.
3. A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal Bases of Compactly Supported Wavelets," *Comm. Pure Appl. Math.,* vol. 45, 1992, pp. 485–560.
4. D. Tay, "Rationalizing the Coefficients of Popular Biorthogonal Wavelet Filters," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 10, no. 6, 2000, pp. 998–1005.
5. A. Pande and J. Zambreno, "Poly-DWT: Polymorphic Wavelet Hardware Support for Dynamic Image Compression," *ACM Trans. Embedded Computing Systems,* vol. 11, no. 1, 2012, article no. 6.
6. D. Engel and A. Uhl, "Parameterized Biorthogonal Wavelet Lifting for Lightweight JPEG 2000 Transparent Encryption," *Proc. ACM 7th Workshop Multimedia and Security* (MM&Sec), ACM, 2005, pp. 63–70.
7. Z. Liu and N. Zheng, "Parametrization Construction of Biorthogonal Wavelet Filter Banks for Image Coding," *Signal, Image and Video Processing,* vol. 1, no. 1, 2007, pp. 63–76.
8. A. Pande and J. Zambreno, "The Secure Wavelet Transform," *J. Real-Time Image Processing,* vol. 7, no. 2, 2012, pp. 131–142.
9. R. Bose and S. Pathak, "A Novel Compression and Encryption Scheme Using Variable Model Arithmetic Coding and Coupled Chaotic System," *IEEE Trans. Circuits and Systems I,* vol. 53, no. 4, 2006, pp. 848–857.
10. G. Jakimoski and K. Subbalakshmi, "Cryptanalysis of Some Multimedia Encryption Schemes," *IEEE Trans. Multimedia,* vol. 10, no. 3, 2008, pp. 330–338.

11. M. Grangetto, E. Magli, and G. Olmo, "Multimedia Selective Encryption by Means of Randomized Arithmetic Coding," *IEEE Trans. Multimedia,* vol. 8, no. 5, 2006, pp. 905–917.
12. H. Kim, J. Wen, and J. Villasenor, "Secure Arithmetic Coding," *IEEE Trans. Signal Processing,* vol. 55, no. 5, 2007, pp. 2263–2272.
13. N. Nagaraj and P.G. Vaidya, "One-Time Pad, Arithmetic Coding and Logic Gates: An Unifying Theme Using Dynamical Systems," *Computing Research Repository,* vol. 803, 2008.
14. A. Pande, P. Mohapatra, and J. Zambreno, "Using Chaotic Maps for Encrypting Image and Video Content," *Proc. IEEE Int'l Symp. of Multimedia,* IEEE CS, 2011, pp. 171–178.

Amit Pande is a project scientist in the Department of Computer Science at the University of California, Davis. His research interests include multimedia coding, communications, encryption, forensics, hardware acceleration, and wireless networks. Pande has a PhD in computer engineering from Iowa State University. He is a member of IEEE. Contact him at amit@cs.ucdavis.edu.

Prasant Mohapatra is the Tim Bucher Family Endowed Chair Professor and the chairman of the Department of Computer Science at the University of California, Davis. His research interests include wireless networks, sensor networks, Internet protocols, and QoS. Mohapatra has a PhD in computer engineering from Penn State University. He is a fellow of IEEE. Contact him at prasant@cs.ucdavis.edu.

Joseph A. Zambreno is an associate professor in the Department of Electrical and Computer Engineering at Iowa State University and the codirector of the Reconfigurable Computing Lab (RCL). His research interests include reconfigurable computing, including the design and implementation of optimized FPGA architectures in various real-world domains such as cryptography, image and video processing, and data mining. Zambreno has a PhD in electrical and computer engineering from Northwestern University. He is a member of IEEE. Contact him at zambreno@iastate.edu.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

ADVERTISER INFORMATION • OCTOBER–DECEMBER 2013

Advertising Personnel

Marian Anderson: Sr. Advertising Coordinator
Email: manderson@computer.org
Phone: +1 714 816 2139 | Fax: +1 714 821 4010

Sandy Brown: Sr. Business Development Mgr.
Email sbrown@computer.org
Phone: +1 714 816 2144 | Fax: +1 714 821 4010

Advertising Sales Representatives (display)

Central, Northwest, Far East:
Eric Kincaid
Email: e.kincaid@computer.org
Phone: +1 214 673 3742
Fax: +1 888 886 8599

Northeast, Midwest, Europe, Middle East:
Ann & David Schissler
Email: a.schissler@computer.org, d.schissler@computer.org
Phone: +1 508 394 4026
Fax: +1 508 394 1707

Southwest, California:
Mike Hughes
Email: mikehughes@computer.org
Phone: +1 805 529 6790

Southeast:
Heather Buonadies
Email: h.buonadies@computer.org
Phone: +1 973 585 7070
Fax: +1 973 585 7071

Advertising Sales Representatives (Classified Line)

Heather Buonadies
Email: h.buonadies@computer.org
Phone: +1 973 304-4123
Fax: +1 973 585 7071

Advertising Sales Representatives (Jobs Board)

Heather Buonadies
Email: h.buonadies@computer.org
Phone: +1 973 304-4123
Fax: +1 973 585 7071