ISSN 1809-9807

The International Journal of

FORENSIC COMPUTER SCIENCE

Volume 4, Number 1, 2009

www.IJoFCS.org

THE INTERNATIONAL CONFERENCE ON CYBER CRIME INVESTIGATION

To register / For more infomation:

www.iccyber.org

ISSN 1809-9807

The International Journal of FORENSIC COMPUTER SCIENCE IJoFCS

www.IJoFCS.org

Volume 4, Number 1, 2009

Brasília, DF - Brazil

ISSN 1809-9807

Cover: Cláudio Miranda de Andrade

SUBSCRIPTION OFFICE

The International Journal of Forensic Computer Science (IJoFCS)

BRAZILIAN ASSOCIATION OF HIGH TECHNOLOGY EXPERTS (ABEAT)
Associação Brasileira de Especialistas em Alta Tecnologia (APCF)
www.abeat.org.br

Address: SCLN 309, Bloco D, Sala 103 - CEP: 70755-540, Brasília/DF, BRAZIL
Phone: +55 61 3202-3006

Web site: www.IJoFCS.org
E-mail: ijofcs@IJoFCS.org

The International Journal of Forensic Computer Science - V. 4, N. 1 (2009) - Brazil
Brazilian Association of High Technology Experts (ABEAT) - Brasilia, Brazil

ISSN 1809-9807

1. Forensic Computer Science

CDD 005.8

The Journal was founded in 2006.

The International Journal of FORENSIC COMPUTER SCIENCE

PUBLISHERS

BRAZILIAN ASSOCIATION OF HIGH TECHNOLOGY EXPERTS (ABEAT)
Associação Brasileira de Especialistas em Alta Tecnologia (APCF)
www.abeat.org.br

Journal's Scope

Biometrics
Computer Crimes
Computer Forensics
Computer Forensics in Education
Computer Law
Criminology
Cryptology
Digital Investigation
Information Security
International Police Cooperation
Intrusion Prevention and Detection
Network Security
Semantic Web

Artificial Intelligence
Artificial Neural Network
Computer Vision
Image Analysis
Image Processing
Machine Learning
Management Issues
Pattern Recognition
Secure Software Development
Signal Processing
Simulation
Software Engineering

e-forensic PRESS

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

Number 1 July 2009

SUMMARY

IJoFCS (2009) 1, 6-7

e-forensic PRESS

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

GUIDE FOR AUTHORS

The *Journal* seeks to publish significant and useful articles dealing with the broad interests of the field of Forensic Computer Science, software systems and services related to Computer Crimes, Computer Forensics, Computer Law, Computer Vision, Criminology, Cryptology, Digital Investigation, Artificial Neural Networks, Biometrics, Image Analysis, Image Processing, International Police Cooperation, Intrusion Prevention and Detection, Machine Learning, Network Security, Pattern Recognition, and Signal Processing. Matters of digital/cyber forensic interest in the social sciences or relating to law enforcement and jurisprudence may also be published.

Our goal is to achieve an editorial balance among technique, theory, practice and commentary, providing a forum for free discussion of Forensic Computer Science problems, solutions, applications and opinions. Contributions are encouraged and may be in the form of articles or letters to the editor.

The *Journal* neither approves nor disapproves, nor does it guarantee the validity or accuracy of any data, claim, opinion, or conclusion presented in either editorial content, articles, letters to the editor or advertisements.

CONTENT

A paper may describe original work, discuss a new technique or application, or present a survey of recent work in a given field. Concepts and underlying principles should be emphasized, with enough background information to orient the reader who is not a specialist in the subject. Each paper should contain one key point, which the author should be able to state in one sentence. The desired focus should be on technology or science, rather than product details. It is important to describe the value of specific work within its broader framework.

Replications of previously published research must contribute sufficient incremental knowledge to warrant publication. Authors should strive to be original, insightful, and theoretically bold; demonstration of a significant "value-added" advance to the field's understanding of an issue or topic is crucial to acceptance for publication. Multiple-study papers that feature diverse methodological approaches may be more likely to make such contributions.

We attach no priorities to subjects for study, nor do we attach greater significance to one methodological style than another. For these reasons, we view all our papers as high-quality contributions to the literature and present them as equals to our readers.

PRESENTATION

A paper is expected to have an abstract that contains 200 words or less, an introduction, a main body, a conclusion, cited references, and brief biographical sketches of the authors. A typical paper is less than 10,000 words and contains five or six figures. A paper should be easy to read and logically organized. Technical terms should be defined and jargon avoided. Acronyms and abbreviations should be spelled out and product names given in full when first used. Trademarks should be clearly identified. References should be numbered sequentially in the order of their appearance in the text.

SUBMISSION INFORMATION

Manuscripts should be submitted in an editable format produced by any word processor (MS Word is preferred). PDF files should be submitted only if there is no alternative.

By submitting a manuscript, the author certifies that it is not under simultaneous consideration by any other publication; that neither the manuscript nor any portion of it is copyrighted; and that it has not been published elsewhere. Exceptions must be noted at the time of submission. Submissions are refereed (double-blind review).

PUBLICATION PROCESS

A submitted paper is initially reviewed to determine whether the topic and treatment are appropriate for readers of the *Journal*. It is then evaluated by three or more independent referees (double-blind review). The policy of double-blind review means that the reviewer and the author do not know the identity of each other. Reviewers will not discuss any manuscript with anyone (other than the Editor) at any time. Should a reviewer have any doubt of his or her ability to be objective, the reviewer will request not to review a submission as soon as possible upon receipt.

After review, comments and suggestions are forwarded to the author, who may be asked to revise the paper. Finally, if accepted for publication, the paper is edited to meet *Journal* standards. Accepted manuscripts are subject to editorial changes made by the Editor. The author is solely responsible for all statements made in his or her work, including changes made by the editor. Proofs are sent to the author for final inspection before publication. Submitted manuscripts are not returned to the author; however, reviewer comments will be furnished.

Reviewers may look for the following in a manuscript:

Theory: Does the paper have a well-articulated theory that provides conceptual insight and guides hypotheses formulation? Equally important, does the study informs or improves our understanding of that theory? Are the concepts clearly defined?

Literature: Does the paper cite appropriate literature and provide proper credit to existing work on the topic? Has the author offered critical references? Does the paper contain an appropriate number of references (e.g., neither over – or under – referencing does not occur)?

Method: Do the sample, measures, methods, observations, procedures, and statistical analyses ensure internal and external validity? Are the statistical procedures used correctly and appropriately? Are the statistics' major assumptions reasonable (i.e., no major violations)?

Integration: Does the empirical study provide a good test of the theory and hypotheses? Is the method chosen (qualitative or quantitative) appropriate for the research question and theory?

Contribution: Does the paper make a new and meaningful contribution to the management literature in terms of all three: theory, empirical knowledge, and management practice?

Citation in a review: Finally, has the author given proper reference or citation to the original source of all information given in their work or in others' work that was cited?

IJoFCS (2009) 1, 8-8

e-forensic PRESS

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

EDITORIAL

Paulo Quintiliano, Ph.D.

Editor-in-chef

The publication of the first issue of Volume 4 of the IJoFCS marks an important step in the development and advancement of Forensic Computer Science. Our journal is taking firm steps to advance its stability and permanence, establishing itself worldwide as a source of new information and knowledge in the area of Forensic Computer Science.

In its four years of existence, the IJoFCS has received high ratings from the scientific bodies that are responsible for classifying the intellectual quality of scientific periodicals in our country. This is surely also—or will soon be—the case in other countries, as copies of our journal spread around the world.

We are continually working towards our objective of bringing together scientists and researchers from every part of the world, in order to develop and advance Forensic Computer Science, and in order to promote a safer cybernetic environment for our society. Each new issue shows the involvement of new researchers, who are interested in and have achieved good results in our subject area, and who join our team as willing collaborators.

In this volume, we feature articles focusing on the following areas: “computer crime”, “computer forensics”, “digital investigation”, “biometrics”, “image processing” and “pattern recognition.”

Once again, we welcome prospective authors to submit their papers to IJoFCS via our webpage at www.ijofcs.org.

IJoFCS (2009) 1, 9-21

e-forensic PRESS

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

Automatic Speaker Recognition with Multi-resolution Gaussian Mixture Models (MR-GMMs)

Frederico Q. D'Almeida(1), Francisco A. O. Nascimento(2), Pedro A. Berger(3), and Lúcio M. da Silva(4)

(1) Brazilian Federal Police -Brazil
(2,4) Department of Electrical Engineering at University of Brasilia - Brazil
(3) Department of Computer Science at University of Brasilia - Brazil

Abstract - Gaussian Mixture Models (GMMs) are the most widely used technique for voice modeling in automatic speaker recognition systems. In this paper, we introduce a variation of the traditional GMM approach that uses models with variable complexity (resolution). Termed Multi-resolution GMMs (MR-GMMs); this new approach yields more than a 50% reduction in the computational costs associated with proper speaker identification, as compared to the traditional GMM approach. We also explore the noise robustness of the new method by investigating MR-GMM performance under noisy audio conditions using a series of practical identification tests.

1. Introduction

Modern automatic speaker recognition (ASR) systems based on Gaussian Mixture Models (GMMs) have proven quite effective at identifying speakers given certain voice segments (Reynolds, 1992). However, high recognition rates require complex models with at least 16 components (Reynolds and Rose, 1995). If a noise-robust system were developed, then the complexity would likely exceed 80 components (D'Almeida *et al.,* 2008; Ming *et al.*, 2007) and carry a proportional increase in computational costs.

In this study, we present a Multi-resolution Gaussian Mixture Model (MR-GMM) speaker recognition technique in which each speaker is represented by at least two distinct models: a low resolution (low complexity) model used to conduct a pre-classification of the speakers, and a high resolution (high complexity) model used to achieve the final classification.

During the first identification stage, a large portion of the modeled speakers are eliminated by submitting the unknown voice signal to all low resolution models. Although such models are too simple to yield a certain match, the process does discard a great majority (up to 85%) of incorrect speakers. During the second identification stage, high resolution models are used to test the best-match results from the first stage. In this manner, calculations using high resolution (high complexity) models are only carried out for a small fraction of the overall number of speakers.

The final result is a two-stage identification process using models of varying levels of complexity (multi-resolution) that yields results similar to traditional GMM systems but at a

considerably lower computational cost. This study also investigates the proposed model's sensitivity to noise by conducting simulations with different noise levels and comparing the results with traditional GMM methods.

2. Gaussian Mixture Model (GMM)

Gaussian Mixture Models are a useful data modeling tool when variables are distinctly clustered (Reynolds, 1992). This distribution is modeled as the weighted sum of *M* Gaussian distributions, each of dimension *D*, as given by

$$p(\vec{x} \mid \lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}). \quad (1)$$

Here $\vec{x}$ is a *D*-dimensional parameter (variable) vector, $b_i(\vec{x})$ are the *M* Gaussian distributions comprising the model, and p_i are the respective weights of each component. Note that *i* ranges from 1 to *M*. Each component of the GMM, $b_i(\vec{x})$, is a *D*-dimensional Gaussian given as

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{(\vec{x}-\vec{\alpha}_i)'\Sigma_i^{-1}(\vec{x}-\vec{\alpha}_i)}{2}\right\}, \quad (2)$$

which has an average value of $\vec{\alpha}_i$ and a covariance matrix Σ_i. The weights of the mixture components are appropriately normalized so that their sum total is unity.

In equation , λ represents the complete description of a GMM including its averages, weights and covariance matrices:

$$\lambda = \{p_i, \vec{\alpha}_i, \Sigma_i\}, i = 1,...,M. \quad (3)$$

In ASR systems, the voice of each speaker is modeled by a different GMM which produces a model λ_s with *s* ranging from 1 to the total number of modeled speakers *S*. The modeled universe of speakers is represented by *U:*

$$U = \{\lambda_s, s = 1,...,S\}. \quad (4)$$

2.1. GMM Training

For GMM training, an audio file is required containing voice recordings of each speaker to be modeled (training files). For each training file, various parameter vectors $\vec{x}_t$ are calculated for different instances in time *t*. The set of these parameter vectors extracted from the training files of a particular speaker is represented by

$$X_s = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_T\}. \quad (5)$$

Note that the speaker index *s* in equation has been removed from the right side of the equation for clarity.

The aim of the GMM training is to adjust the parameters of the model of speaker, λ_s, in order to maximize the probability of occurrence of the set of parameter vectors X_s. To simplify the problem, it is assumed that each parameter vector $\vec{x}_t$ is independent of the others, allowing the following notation:

$$p(X_s \mid \lambda_s) = \prod_{t=1}^{T} p(\vec{x}_t \mid \lambda_s). \quad (6)$$

This is a non-linear function of the parameters of model λ_s, which does not allow direct maximization. Generally, maximization of is performed with the *Expectation-Maximization* (EM) algorithm as described by Dempster *et al.* (1977).

2.2. Speaker Identification

To identify the speaker belonging to the test voice file, one must determine which of the models λ_s, of universe *U*, present the greatest *a posteriori* probability of a set of parameters calculated from the given test file. That is,

$$\tilde{s} = \arg\max_{\lambda_s \in U} p(\lambda_s \mid Y) = \arg\max_{\lambda_s \in U} \frac{p(Y \mid \lambda_s)\, p(\lambda_s)}{p(Y)}, \quad (7)$$

where *Y* is the set of parameter vectors calculated from the test file, $Y = \{\vec{y}_1, \vec{y}_2, ..., \vec{y}_T\}$, and Bayes' rule is applied.

Assuming that all speakers are equally probable, $p(\lambda_s) = cte., s = 1, ..., S$, and considering that *p(Y)* is a constant depending solely on the tested recording (and therefore the same for all speakers in the universe), then identifying the speaker is as simple as calculating

$$\tilde{s} = \arg\max_{\lambda_s \in Y} p(Y \mid \lambda_s). \quad (8)$$

If one also assumes independence among the elements of the test parameter vector, as formulated in for the training parameters, and maximize the logarithm of the probability instead, then equation becomes

$$\tilde{s} = \arg\max_{\lambda_s \in U} \sum_{t=1}^{T} \log p(\vec{y}_t \mid \lambda_s). \quad (9)$$

Note that using the logarithm helps to avoid numerical problems since the probabilities involved in equation are extremely small.

Since the length of times for each speaker's audio file are not exactly equal, equation is normalized with respect to time as

$$\tilde{s} = \arg\max_{\lambda_s \in U} \frac{\sum_{t=1}^{T} \log p(\vec{y}_t \mid \lambda_s)}{T}. \quad (10)$$

It is assumed that correct identification takes place when the speaker who maximizes equation , $\tilde{s}$, is in fact the correct speaker $\hat{s}$:

$$\tilde{s} = \hat{s}. \quad (11)$$

2.3. Computational Cost

The computational cost associated with identifying the speaker is a function of several factors. For example, it depends on the number of speakers in the universe *S*, since all models need to be simulated to find the specific model that maximizes equation ; the duration of the test file used since the voice parameter vectors $\vec{y}_t$ are extracted at fixed time intervals; the dimension of the Gaussian components *D* used in the model; and the number of components *M* of the models

All parameters were studied when minimizing the computational cost associated with speaker identification except the number of speakers in the universe, which cannot be altered for a given application (Reynolds and Rose, 1995). For the number *M* of model components, Reynolds and Rose (1995) determined that at least 8 to 16 components are required for good system performance with noiseless audio; a result confirmed during this study as well. When developing multi-conditional systems robust to noise, the minimum number of components increases to between 64 and 128 according to D'Almeida *et al.* (2008) and Ming *et al.* (2007).

From the definition of the GMM in equation , the total computational cost *W* of an identification task (during the test phase only) is approximately proportional to the number of Gaussians *M* in the speaker models λ. In reality, for each new component introduced into the models, it is necessary to calculate a new Gaussian defined by equation , multiply it by the coefficient of the mixture, and then add the new product to the renormalization sum in equation . The computational costs for temporal normalization and maximum identification (the *argmax* function) expressed in equation are independent of the number of model components. These costs are of little overall significance since the calculations are executed only once for each speaker and test file, as compared to the calculations for equations and which are executed once for each parameter vector y_i. Thus, even for short test files lasting only 1 second, there are still 50 calculations carried out for equations and when evaluating equation . The cost for calculating the logarithm is also independent of the number of model components, and even though it must be performed for each parameter vector y_i, it is still of little relevance since it consists of a single scalar operation.

We also consider the cost of extracting the parameters y_i from the questioned audio file. The parameters which are most frequently used in ASR systems are the Mel Frequency Cepstral Coefficients (MFCC) which offer the best identification performance (D'Almeida and Nascimento, 2006). In this case, FFT and other calculations are needed, which carry relatively high computational complexities. However, since this calculation is carried out just once for the whole procedure, then this cost will not be as significant in terms of the total identification effort with a sufficiently large speaker universe.

Therefore, under certain conditions normally met by ASR systems, it can be stated with certain precision that the identification cost for GMM models is proportional to the quantity of model components:

$$W_{GMM} \propto M \quad . \tag{12}$$

3. Multi-Resolution Gaussian Mixture Models (MR-GMM)

There have been several experiments aimed at minimizing the number of GMM components without compromising speaker identification (e.g., Reynolds and Rose, 1995), but one issue that has not been fully addressed in the literature is how optimizing models with less than 16 components (for noiseless audio) significantly reduces the performance of the system in terms of the number of correctly identified speakers. However, in this situation, it is expected that the correct model still receives one of the highest scores (on equation (10) classification). Thus, modeling with a fewer number of components may not be capable of exactly determining the correct speaker, but must be capable of separating, within the universe, a subgroup containing the correct speaker.

To take advantage of this idea, loosening the success condition might allow high positive identification rates even for models with only 2 or 4 components. This can be expressed mathematically as:

$$\hat{s} \in \tilde{U} \ , \tag{13}$$

where $\tilde{U}$ is a subset of the speaker universe *U* given by

$$\tilde{U} = \left\{ \lambda_s \in U, \left[\frac{\sum_{t=1}^{T} \log p\left(\vec{y}_t \mid \lambda_s\right)}{T} \right] \geq \xi \right\}. \tag{14}$$

Here ξ is the *C*-th greatest value of the expression between brackets in equation calculated for all models in the universe. The value of *C* is defined based on the model order used and the required performance.

Of course, loosening the system's success condition as described by equations and leads to a new problem: the resulting identification is no longer a single speaker, but instead a set of *C* speakers. Naturally, this does not produce a precise speaker identification. However, the precise identification can then be determined through a new GMM system, now with 16 components, and using a standard identification process as expressed in equation and applied to the restricted set of speakers $\tilde{U}$.

In this study, we propose a systematic speaker identification process in successive stages through GMM models of various resolutions (number of components). This new modeling approach is termed Multi-resolution Gaussian Mixture Models (MR-GMMs).

3.1. Mathematic Formulation and Training

The MR-GMMs are essentially extensions of the traditional GMM approach. The main difference between these two techniques is that, for a given speaker, the MR-GMM approach has two or more distinct GMM models with different degrees of complexity (number of components). Analogous to equation (3), the MR-GMM approach may be formulated as

$$\Lambda=\{\lambda_k, k=1,...,K\}=\left\{\begin{matrix}\{p_{1,i_1},\vec{\alpha}_{1,i_1},\Sigma_{1,i_1}\}\\ \{p_{2,i_2},\vec{\alpha}_{2,i_2},\Sigma_{2,i_2}\}\\ \vdots\\ \{p_{K,i_K},\vec{\alpha}_{K,i_K},\Sigma_{K,i_K}\}\end{matrix}\right\}\begin{matrix}i_1=1,...,M_1;\\ i_2=1,...,M_2;\\ \vdots\\ i_K=1,...,M_K\end{matrix}, \quad (15)$$

where $M_k > M_{k-1}$. Note that the subscript k for model λ_k does not index different speakers as index s does in equation . Rather, k refers to the submodels comprising a single MR-GMM. Consequently, all models λ_k are of a single speaker. The set of all MR-GMM models or the universe of modeled speakers U is defined similar to equation as

$$U=\{\Lambda_s, s=1,...,S\}. \quad (16)$$

The training for each submodel λ_k of a single MR-GMM model Λ is carried out as the training of a normal and independent GMM model. The different submodels of a single speaker may be trained from the same audio segment; this has no effect on the global model since the aim of the MR-GMM is to use models of different resolutions to minimize the overall computational cost during speaker identification. In reality, training the submodels with the same audio segment is a more natural alternative since it eliminates the need to alter the existing databases.

3.2. Speaker Identification

There is a significant difference between the MR-GMM and GMM approaches during the speaker identification phase (test phase). For GMMs, the model of each speaker is evaluated according to the tested audio segment in order to find the most likely match according to equation . On the other hand, the MR-GMM approach does not conduct the test in a single evaluation. Successive test stages are carried out, each using a model with a resolution greater than its predecessor, and speakers are gradually selected until the best candidate is determined in the final stage.

The fundamental idea behind MR-GMMs is that they can reduce the computational cost of identifying the speaker by reducing the average complexity of the models used while not sacrificing overall performance. This requires a gradual speaker selection process using models of increasing complexity, using high-complexity models only for a limited number of speakers in the universe.

During the first test phase, less complex models for each speaker, $\Lambda_{s,1}$, are first used to select the C_1 models from universe U which best matches the test audio. Note that we use $\Lambda_{s,k}$ as the GMM submodel λ_k associated with the MR-GMM Λ_s. The result of the first test phase is a subset of the speaker universe, denoted as U_1, containing C_1 models of the best-matching speakers at low resolution:

$$U_1=\left\{\Lambda_s\in U, \left[\frac{\sum_{t=1}^{T}\log p\left(\vec{x}_t\mid\Lambda_{s,1}\right)}{T}\right]\geq\xi_1\right\}, \quad (17)$$

where ξ_1 is the C_1-th greatest value of the expression between brackets in equation , evaluated for all $\Lambda_{s,1}$ models in U.

During the second phase of the test, the initial procedure is repeated but now with models with the second lowest resolution $\Lambda_{s,2}$ in universe U_1 from the previous stage. In this phase, the result is a subset $U_2\subset U_1$ given by

$$U_2=\left\{\Lambda_s\in U_1, \left[\frac{\sum_{t=1}^{T}\log p\left(\vec{x}_t\mid\Lambda_{s,2}\right)}{T}\right]\geq\xi_2\right\}, \quad (18)$$

where ξ_2 is the C_2-th greatest value of the expression in brackets evaluated for models $\Lambda_{s,2}$, $s\in U_1$.

The testing process gradually reduces the speaker universe according to

$$U_{k+1} = \left\{ \Lambda_s \in U_k, \left[\frac{\sum_{t=1}^{T} \log p\left(\overline{x}_t \mid \Lambda_{s,k+1}\right)}{T} \right] \geq \xi_{k+1} \right\}, \quad (19)$$

until the last stage *K* where the model that best adjusts to the audio test (thus, where C_K is always equal to 1) is determined by the expression analogous to equ

$$\hat{S} = U_K = \left\{ \Lambda_s \in U_{k-1}, \frac{\sum_{t=1}^{T} \log p\left(\overline{x}_t \mid \Lambda_{s,K}\right)}{T} \geq \xi_K \right\} = \arg\max_{s \in U_{k-1}} \frac{\sum_{t=1}^{T} \log p\left(\overline{x}_t \mid \Lambda_{s,K}\right)}{T}. \quad (20)$$

3.3. Computational Advantage

The computational advantage of using MR-GMM is from the possibility of reducing the average complexity of the models used during speaker identification. Thus, it is essential to determine the parameters M_k, the number of components of each of the submodels $\Lambda_{s,k}$, and the quantity of speakers classified to the next stage C_k.

The total computational cost for speaker identification with MR-GMM is given by

$$W_{MR-GMM} \propto M_1 + M_2 \frac{C_1}{S} + \ldots + M_K \frac{C_{K-1}}{S} = \sum_{k=1}^{K} M_k \frac{C_{k-}}{S}, \quad (21)$$

where, for simplicity, we define $C_0=S$, indicating that all speakers are considered in the first evaluation. Note that since more than one test is conducted for each speaker (speakers tested and classified in stage *k* are tested again in stage *k+1*), a poor choice of M_k and C_k may cause an increase in the total computational cost compared to traditional GMM models. For example, assuming a MR-GMM model with only two resolutions M_1=8 and M_2=16 (that is, the first phase of the test is conducted with an 8-component model, and the second and last phase is conducted with a 16-component model) and with $C_1=S/2$ (50% of the best models in the universe pass on to the second training phase), then the total cost *W* of the process is

$$W_{MR-GMM} \propto M_1 + M_2 \frac{C_1}{S} = 16 = M_2 = W_{GMM}, \quad (22)$$

and thus there would be no reduction in the computational cost over a 16-component GMM model.

In addition, as long as the values of M_k and C_k are adequately adjusted, one expects that a MR-GMM model conducting its last cycle of tests comprised of M_K components would have a speaker identification performance equivalent to a GMM model with M_K components. For this reason, computational cost comparisons must be made using a GMM model of this complexity.

In order to effectively reduce the total computational cost of the process, parameters M_k and C_k must be selected such that

$$W_{MR-GMM} \propto \sum_{k=1}^{K} M_k \frac{C_{k-1}}{S} < M_K = W_{GMM}. \quad (23)$$

The relative reduction of the computational cost of the identification process can then be calculated as

$$G = 1 - \frac{W_{MR-GMM}}{W_{GMM}} = 1 - \frac{\sum_{k=1}^{K} M_k \frac{C_{k-1}}{S}}{M_K} = 1 - \sum_{k=1}^{K} \frac{M_k}{M_K} \frac{C_{k-1}}{S}. \quad (24)$$

4. Performance Evaluation and Results

Simulations used to analyze the performance of MR-GMM models were conducted using the voice database described by D'Almeida *et al.* (2007). The characteristics are this dataset are described below.

4.1. Description of the Audio Database

The voice database consisted of 30 different speakers (*S* = 30), half male and half female. Each speaker was recorded while reading identical pre-defined texts, and each recording was broken into 21 files with the same starting and ending points

for all speakers. Thus 21 files were generated for each speaker, indicated by $A_{n,s}$ where *s* indicates the speaker and *n* indicates the segment of the recorded file. The first segment of the 21 audio clips for each speaker was used to train the MR-GMM models, and the remaining 20 segments were used to carry out the 20 different identification tests.

All recordings were made in acoustically prepared environments with professional microphones and audio capture cards. Files were acquired at a sampling rate of 22 kHz, 16-bit quantization, in monaural mode. From this initial database, versions of the audio files were generated with other sampling frequencies and codifications: 16 kHz/16 bits, 11 kHz/16 bits, 11 kHz/8 bits-∞-law, 8 kHz/8 bits-∞-law, and 8 kHz/8 bits. Identification tests were conducted using all of the above file versions.

Before carrying out the tests, all audio files were normalized such that their peak amplitude corresponded to 100% of the maximum quantization value. Silent segments were then excluded from the files by an automatic silence detector based on the signal energy measured in 20 ms windows using a 15 ms window overlap (thus 5 ms increments) and a manually defined silence threshold (a single value for all files) based on practical tests.

For all simulations, MFCC parameters were used as the modeling parameters since this has produced the best results in ASR applications (D'Almeida and Nascimento, 2006). The parameters were calculated for each 20 ms audio window with no overlapping, using filter banks applied directly to the signal frequency spectrum calculated in the same window. From each window, 12 MFCC parameters were extracted. Parameter normalization was performed for both the model and test phases as outlined in Reynolds and Rose (1995), as a way of increasing system performance (removal of the average values of the coefficients).

4.2. TEST PROCEDURE AND RESULTS

Tests were carried out by comparing the performances (number of correct identifications) of a traditional 16-component GMM system ($W_{GMM} = 16$) with three MR-GMM systems. All MR-GMM models were constructed with only two identification stages (resolution) ($K = 2$), given the relatively small speaker universe available in the database (S = 30). Diagonal covariance matrixes Σ_i were used for all models since Reynolds and Rose (1995) demonstrated that this has no negative effect on the overall performance.

The first MR-GMM model, called MR-GMM 1, used parameters $C_0=30$, $C_1=5$, $M_1=6$, and $M_2=16$. The computational cost of this model calculated using equation is

$$W_{MR-GMM\,1} \propto \sum_{k=1}^{K} M_k \frac{C_{k-1}}{S} = 6+16\frac{5}{30} = 8.67 \quad (25)$$

The second MR-GMM model, called MR-GMM 2, used parameters $C_0=30$, $C_1=5$, $M_1=4$, and $M_2=16$ and produced a computational cost of

$$W_{MR-GMM\,2} \propto \sum_{k=1}^{K} M_k \frac{C_{k-1}}{S} = 4+16\frac{5}{30} = 6.67 \quad (26)$$

The third MR-GMM model, called MR-GMM 3, used parameters $C_0=30$, $C_1=5$, $M_1=2$, and $M_2=16$ and resulted in a computational cost of

$$W_{MR-GMM\,3} \propto \sum_{k=1}^{K} M_k \frac{C_{k-1}}{S} = 2+16\frac{5}{30} = 4.67 \quad (27)$$

The reductions in the computational costs of the MR-GMM models, as calculated by equation , are 45.81%, 58.33% and 70.83%, respectively.

To carry out the comparisons in similar conditions and exclude affects other than the difference in modeling techniques, the submodels of the 16-component MR-GMM ($\Lambda_{s,2}$) were the same for all MR-GMM models and were also used as the GMM models of each speaker (λ_s). Thus, it was possible to eliminate performance differences from the training of the models so that the alterations in the correct identification rates only reflected the difference between the MR-GMM modeling techniques using progressive identification stages versus the traditional GMM technique.

The results of the tests are presented in Table 1. Note that the MR-GMM 1 and MR-GMM 2 models gave results that closely matched the traditional GMM model, while MR-GMM 3 suffered from significant performance loss. Thus MR-GMM models can provide a computational cost reduction of up to 58% with no relevant losses in system performance for noiseless audio. Also note that for the MR-GMM 1 and MR-GMM 2 models, significant performance losses were only observed for the 8 kHz audio sampling frequency and 8 bit codification. For all other cases, there was no significant performance alteration since the maximum difference in correct identification rates was limited to 0.3 percentage points for the MR-GMM 1 model and to 0.5 percentage points for the MR-GMM 2 model.

4.3. Robustness to noise

To analyze the noise-sensitivity of the MR-GMM models, simulations were performed using noiseless audio samples and with the same samples after adding various levels of noise. In these simulations, only the MR-GMM 1 and MR-GMM 2 models were used since the performance of the MR-GMM 3 model was considered unsatisfactory even for noiseless audio.

The noisy audio samples were generated from the "noiseless" audio (the originally captured files) by adding white noise to the files. Even though the noise presence was simulated, practical tests with ASR systems carried out by Ming *et al.* (2007) confirmed that the computational addition of noise is quite similar to the physical (acoustic) addition of noise present during the moment of capture.

The values of the signal-to-noise ratios (SNRs) used in the simulations were 60 dB, 50 dB, 40 dB, 30 dB, 26 dB, 20 dB, 16 dB, 10 dB, 8 dB and 5 dB. The 60 db SNR value corresponds to the audio acquisition system's intrinsic SNR; this value was estimated from the average energy of the signal during the moments of silence (absence of voice) and during speaking. Thus, no additional noise was inserted for the 60 dB SNR audio. For the remaining cases, noise addition was performed so as to maintain a particular average SNR over the entire audio segment. This was done by calculating the average energy of the signal, E_s, in the audio sample according to

$$E_s = \sum_m y_s^2[m], \tag{28}$$

where y_s is the audio signal vector and m the temporal index of the samples. A noise vector y_n was then generated containing samples with zero mean and Gaussian-distributed amplitudes, and having the same dimension of the signal vector y_s. The energy of the noise vector was then calculated as

$$E_n = \sum_m y_n^2[m]. \tag{29}$$

Subsequently, the amplitudes of the noise vector were adjusted so as to obtain the desired SNR, that is

$$y'_n = y_n . SNR . \sqrt{\frac{E_s}{E_n}}. \tag{30}$$

Finally, the noise vector with adjusted amplitudes was added to the original signal vector, generating a audio vector y used in the analysis:

$$y = y_s + y'_n. \tag{31}$$

For the noise robustness analyses, only the following sampling frequencies and codifications were used: 22 kHz / 16 bits, 11 kHz / 16 bits, 8 kHz / 8 bits ∞-law, and 8 kHz / 8 bits linear.

It must be pointed out that calculating the noiseless audio energy was done after removing the silent segments, since this removal was part of the signal pre-processing. Thus, the average power (total energy per total time) calculated from this signal is greater than it would be if the entire signal (including the silent excerpts) were considered. Consequently, to obtain an established SNR, the average power of the noise signal to be

added to the signal, y'_n, is likewise greater than it would be for the entire audio file (including silent segments). For the conducted tests, it was found that the SNR values in the present study are equal, on average, to values nearly 10% greater than those that would be obtained if the noise addition process were conducted on the entire audio file.

Note that the SNR measurement methodology used in this study was chosen for several reasons. First, since the speaking rhythm and pause intervals between words and sentences varied according to each individual, the measurement of the global SNR would depend on these periods of silence. Thus, different SNR values would be obtained even if the average power of the signal (in the segments with voice) and the noise remained fixed. Second, the detection of the speech and silence segments is much simpler for noiseless audio files for which a simple energy detector may be used.

The MR-GMM results obtained using the noise-augmented files were compared to results obtained from traditional GMM results using the same files. Just as in the noiseless audio analysis, the submodels of the 16-component MR-GMM, $\Lambda_{s,2}$, were used as GMM models of each speaker, λ_s, in order to eliminate performance differences resulting from the training of the models. The results of the tests conducted with models MR-GMM 1 and MR-GMM 2 are organized in Tables 2 through 5 below. Visually summaries of Tables 2 through 5 are presented in Figures 1 through 4 as well.

For the 22 kHz audio files, the performance differences between the MR-GMM and GMM approaches were extremely small. For MR-GMM 2, this difference was limited to 2.2 percentage points for the worst case and had an average difference of 0.7 percentage points. For the MR-GMM 1 model, the differences were limited to 0.2 percentage points for the worst case and the model performed on average as well as the traditional GMM approach.

For the 11 kHz audio files, the MR-GMM 2 model again had a worst-case performance degradation of 2.2 percentage points and an average degradation of 0.4 percentage points. The MR-GMM 1 model had no degradation in this simulation.

Simulations using the 8 kHz $\propto$-law audio indicate that the MR-GMM 2 approach has a degradation limited to 1.2 percentage points and an average difference of 0.4 percentage points. The MR-GMM 1 model has a maximum degradation of 0.5 percentage points but performed on the average 0.2 percentage points better than the traditional GMM approach.

Results for the 8 kHz 8-bit linear audio files show a maximum loss of 3.2 percentage points and an average loss of 0.6 percentage points for the MR-GMM 2 model, and a maximum of 1.2 percentage points and average of 0.1 percentage points for the MR-GMM 1 model.

5. Conclusion

The Multi-resolution Gaussian Mixture Models, or MR-GMMs, proposed in this study proved to be an effective alternative for developing high-performance automatic speaker recognition systems with significant reductions in computational costs over traditional Gaussian Mixture Models. Our simulations indicate that cost reductions of up to 58.3% for noiseless audio are possible with no significant degradation in the speaker identification performance. For noisy audio, depending on the sampling frequency and type of codification, reductions of as much as 45% to 58% are possible with no relevant losses in the identification rates as compared to the traditional GMM approach. Note that these results hinge on a relatively small sample database of 30 speakers, thus the use of MR-GMM models with more than two identification stages was not feasible. This may have limited the gain in computational cost, and larger databases using more identification stages may yield further improvements in computational efficiencies.

REFERENCES:

[1] D'Almeida. F.Q. e Nascimento. F.A.O. (2006). Comparação de Desempenho de Parâm. da Fala em Sist. de Rec. Auto. de Locutor. Congresso Brasileiro de Automática – CBA 2006.

[2] D'Almeida. F.Q.. Nascimento. F.A.O. Berger. P.A.. da Silva. L. M. (2007). Efeitos da Codificação MP3 em Sist. de Rec. Auto. de Locutor via GMM. XXV Simpósio Brasileiro de Telecomunicações – SBrT 2007.

[3] Dempster. A.. Laird. N. e Rubin. D. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal Royal Statistical Society. Vol. 39. pp. 1-38.

[4] Ming. J.. Hazen. T.. Glass. J.R. e Reynolds. D.A. (2007). Robust Speaker Recognition in Noisy Conditions. IEEE Trans. on Audio. Speech and Lang. Proc.. vol. 15. pp. 1711-1723.

[5] Reynolds. D.A. (1992). A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification. Ph. D. Thesis. Georgia Inst. of Tech.

[6] Reynolds. D.A. e Rose. R.C. (1995). Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Trans. Speech and Audio Proc.. Vol. 3. no. 1. pp 72-83.

TABLE 1:
Correct identification rates (%) and computational cost reducti

Sampling Freq. Codification	Model GMM	MR-GMM 1	MR-GMM 2	MR-GMM 3
22 kHz / 16 bits	100.0	100.0	100.0	99.2
16 kHz / 16 bits	100.0	100.0	100.0	99.0
11 kHz / 16 bits	99.8	99.5	99.3	91.7
11 kHz / 8 bits ∝-law	95.7	95.7	95.2	91.0
8 kHz / 8 bits ∝-law	96.2	96.2	96.2	93.8
8 kHz / 8 bits	98.7	97.5	96.8	89.8
Cost reduction (%)	-	45.8	58.3	70.8

Table 2:
Correct identification rates (%) for 22 kHz / 16 bits audio

Noise Test (dB)	Model GMM	MR-GMM-1	MR-GMM-2
60	100.0	100.0	100.0
50	100.0	100.0	100.0
40	99.3	99.3	99.3
30	93.7	93.7	93.7
26	83.0	83.2	81.8
20	54.7	54.7	52.5
16	28.5	28.8	26.5
10	13.8	13.7	13.2
8	12.0	12.0	11.3
5	10.0	10.0	10.2

Table 3:
Correct identification rates (%) for 11 kHz / 16 bits audio

Noise Test (dB)	Model GMM	MR-GMM-1	MR-GMM-2
60	99.8	99.8	99.8
50	99.8	99.8	99.8
40	98.5	98.5	98.5
30	94.3	94.3	92.2
26	83.5	83.5	83.5
20	53.7	53.8	52.0
16	30.2	30.2	30.2
10	10.7	10.7	10.7
8	6.0	6.0	6.0
5	2.8	2.8	2.8

Table 4:
Correct identification rates (%) for 8 kHz / 8 bits $\propto$-law audio

Noise Test (dB)	Model GMM	MR-GMM-1	MR-GMM-2
60	99.0	99.0	98.5
50	99.2	99.2	98.0
40	98.2	98.2	97.5
30	93.3	93.5	92.2
26	81.5	81.0	80.5
20	48.3	48.8	48.5
16	30.2	31.2	30.7
10	14.0	13.8	13.2
8	9.3	9.7	9.3
5	5.8	6.3	6.0

Table 5:
Correct identification rates (%) for 8 kHz / 8 bits

Noise Test (dB)	Model GMM	MR-GMM-1	MR-GMM-2
60	95.2	95.2	95.2
50	96.2	96.2	96.2
40	97.2	97.2	97.2
30	96.5	96.5	96.0
26	93.5	93.5	93.2
20	71.7	71.7	69.3
16	46.2	45.0	43.0
10	11.5	11.5	11.7
8	8.3	8.5	8.5
5	5.7	5.7	5.3

FigurE 1:
Correct identification rates
for 22 kHz / 16 bits audio

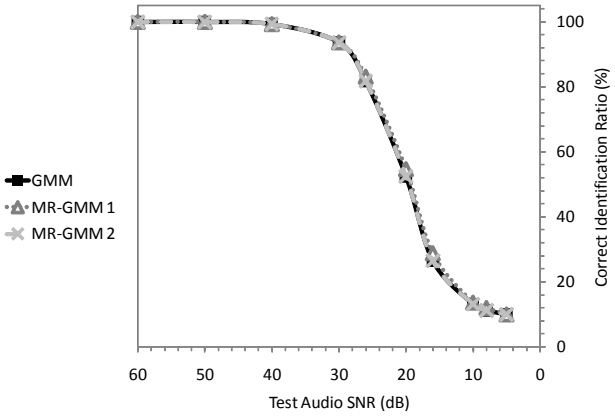


FigurE 3:
Correct identification rates
for 8 kHz / 8 bits ∝-law audio

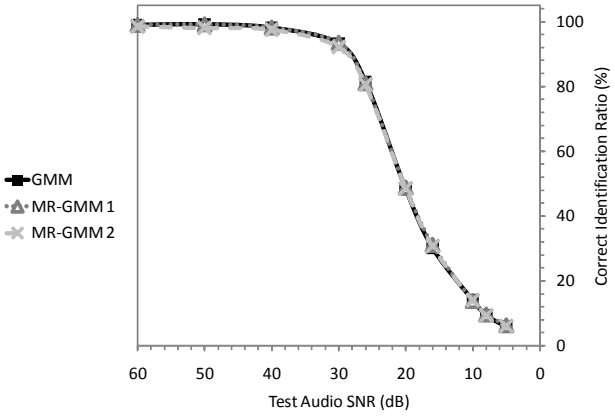


FigurE 2:
Correct identification rates
for 11 kHz / 16 bits audio

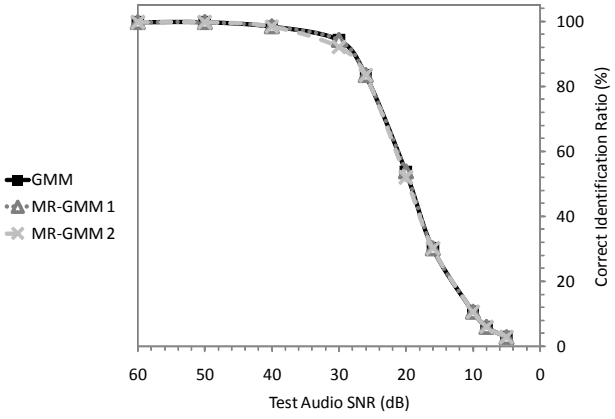
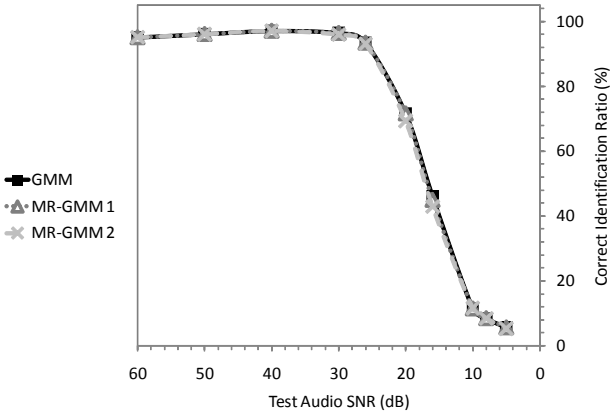


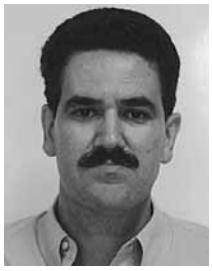
FIGURE 4:
Correct identification rates
for 8 kHz / 8 bits



F.Q.D'Almeida. He was born in Salvador-BA, Brazil, on January 24, 1978. He graduated in Electrical Engineering, Federal University of Bahia - UFBA, Salvador-BA, Brazil, 2000, got his Master Degree in Electrical Engineering, UFBA, 2003, and graduated in Physics, University of Brasilia - UnB, 2006. He is pursuing his Doctorate Degree in Electrical Engineering at UnB. His field of study is Automatic Speaker Recognition. He also works as a forensic expert at Brazilian Federal Police.

F. Assis Nascimento received his B.Sc. in Electrical Engineering from the University of Brasilia in 1982, his M.Sc. in Electrical Engineering from the Federal University of Rio de Janeiro (UFRJ), in 1985, and his Ph.D. in Electrical Engineering from UFRJ in 1988. Currently, he is an Associate Professor at the University of Brasilia and a coordinator of the GPDS (Grupo de Processamento Digital de Sinais).

Pedro de A. Berger graduated in Electrical Engineering at Federal University of Ceara - UFC, Fortaleza-CE, Brazil in 1999, earned his M.Sc. in Electrical Engineering at the University of Brasília (UnB) in 2002 and his Ph.D. at UnB in 2006. He has been a professor in the Department of Computer Science at UnB since 2006. His field of study includes digital signal processing, artificial neural networks and biomedical engineering.



Lúcio M. da Silva was born in Delfinópolis, MG, Brazil, on April 27, 1958. He received the B.S. in electrical engineering from Pontifical Catholic University of Minas Gerais, in 1981, the M.S. degree from University of Brasilia, in 1989, and the Ph. D. degree from Pontifical Catholic University of Rio de Janeiro, in 1996. He is with the Electrical Engineering Department of University of Brasilia. He is involved in teaching and research activities in speech signal processing and digital transmission systems.

IJoFCS (2009) 1, 22-28

e-forensic PRESS

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

Classification of Instant Messaging Communications for Forensics Analysis

Angela Orebaugh(1) , and Jeremy Allnutt(2)

(1) George Mason University, USA, angela_orebaugh@yahoo.com
(2) George Mason University, USA, jallnutt@ece.gmu.edu

Abstract - Instant messaging (IM) is a well-established means of fast and effective communication. Once used primarily by home users for personal communications, IM solutions are now being deployed by organizations to provide convenient internal communication. This often includes the exchange and discussion of proprietary and sensitive information, thus introducing privacy concerns. Although IM is used in many legitimate activities for conversations and message exchange, it can also be misused by various means. For example, an attacker may masquerade as another user by hijacking the connection, performing a man-in-the-middle attack, or by obtaining physical access to a user's computer. There are various reasons that an attacker might want to masquerade as someone else, including spying, disgruntlement, snooping, or other malicious intentions. Analysis of IM in terms of computer forensics and intrusion detection has gone largely unexplored until now. This paper explores IM author classification based on author behavior. Author classification may be used for author identification/validation for forensics analysis or masquerade detection. The experiments presented here applied classification methods to IM messages to determine whether the author of an IM conversation could be identified based strictly on user behavior, and to determine the strongest identifying characteristics.

1. Introduction

Author identification, also called authorship attribution, is the task of determining the author of a piece of work. All humans have unique patterns of behavior, much like the uniqueness of biometric data. Therefore, certain characteristics pertaining to language, composition, and writing, such as particular syntactic and structural layout traits, patterns of vocabulary usage, unusual language usage, and stylistic traits, should remain relatively constant. The identification and learning of these characteristics with a sufficiently high accuracy is the principal challenge in author identification.

This paper describes the methods, analysis, and results of using data mining classification to perform author identification on instant messaging (IM) communications for the purpose of computer forensics analysis. The classification methods presented here profiled author behavior using various linguistics patterns and characteristics. The principle objectives were to create a set of characteristics that remained relatively constant for a large

number of messages written by a single author, and to classify these messages as belonging to a particular author.

The goals of this research were to answer the following questions:

- *Can we identify an author of an IM conversation based strictly on author behavior?*
- *What behavior characteristics are the strongest classifiers?*

This research investigated the foundational techniques necessary to incorporate IM author identification into computer forensics investigations (as digital evidence) and intrusion detection technologies (such as masquerade detection).

2. Related Work

The area of IM communications has been largely unexplored thus far. There has been significant research in identifying authors of text, such as Shakespeare's works and the Federalists papers, as well as in email and virus authorship identification. There has also been some research in online user behavior. However, the results of these research areas have not been applied to IM. Some important works related to IM communications research include the following:

- *Can Online Behavior Unveil Deceivers?* by L. Zhow and D. Zhang. This research explores online deception behavior in a group IM setting.
- *Extracting Social Networks from Instant Messaging Populations*, by J. Resig, S. Dawara, C. M. Homan, and A. Teredesai. This research investigates IM communities as social networks.
- *A Framework for Mining Instant Messaging Services*, by J. Resig and A. Teredesai. This research explores various data mining issues and how they relate to IM and counter-terrorism efforts.
- *Applying Authorship Analysis in Cybercrime Investigation*, by R. Zheng, Y. Qin, Z. Huang, and H. Chen. This research presents an authorship analysis approach for identity tracing in cybercrime investigations.
- *Multi-Topic E-mail Authorship Attribution Forensics*, by O. de Vel, A. Anderson, M. Corney and G. Mohay. This research investigates the forensics authorship identification or categorization of multi-topic e-mail documents.
- *Language and Gender Author Cohort Analysis of E-mail for Computer Forensics*, by O. de Vel, A. Anderson, M. Corney, and G. Mohay. This research investigates the gender and language background of an author, based on cohort attribution mining from e-mail text documents.
- *Gender-Preferential Text Mining of E-mail Discourse*, by O. de Vel, A. Anderson, M. Corney, and G. Mohay. This study provides additional research on gender identification, based on user text analysis.

The first three works are related to IM, but are not focused on author identification. The last four works focus on applying user behavior and pattern analysis techniques to email, but do not consider IM. It is a natural extension to apply the techniques used for email, forensics, and other purposes to IM author profiling and identification.

3. IM Architecture

Most IM networks use a client-server model in which a service provider maintains the server. Users register themselves with the service provider, and download a compatible client for use on the service provider network. Users connect to the central server with the client and begin adding and conversing with other network users, commonly designated as "buddies" or "friends". Buddies are maintained in a Buddy List, which shows when users are logged on for communication. Popular IM service provider networks include AOL, Yahoo,

Table 1
Instant Messaging Author Behavior Categories

Stylometric Features
Character frequency distribution (upper/lowercase, numbers, and special characters)
Word frequency distribution
Emoticon frequency distribution
Function word frequency distribution
Short word frequency distribution
Punctuation frequency distribution
Average word length
Average words per sentence
Contains a greeting
Contains a farewell
Abbreviation frequency distribution
Spelling errors
Grammatical errors

MSN, and Google. Each of these networks provides a compatible communication client. Some clients, such as Trillian and Gaim, can connect to multiple provider networks at once. Most client products allow logging of IM conversations. The IM conversations are logged in a simple text format, making it easy to parse and analyze conversation data. This paper used IM conversation logs for author identification and validation.

4. Author Behavior Categorization

Author behavior categorization uses a set of characteristics that remain relatively constant for a large number of IM messages written by an author. These characteristics, known as *stylometric features*, include syntactic and structural layout traits, patterns of vocabulary usage, unusual language usage, and stylistic features. Each author has various stylometric features that are sufficient to uniquely identify him or her. Stylometric features are often word-based, including word and character frequency distributions, word length, and sentence length. Literary analysts and computational linguists often use frequency lists. Various syntactic features are also included, such as the use of function words (short all-purpose words such as "the" and "to"), punctuation, greetings and farewells, and emoticons. Users also use abbreviations for common phrases such as LOL (laughing out loud) and ROTFL (rolling on the floor laughing), as well as shortened spellings of

Table 2
Pre-defined Attributes

Category	Attribute
Special characters	. , ! ? @ # $ % ^ & * - _ + = ‘ \
Emoticons	:-) :) :-(:(;-) ;) :-P :P ;-P ;P :-D :D :’-(:’(:* :-*
Abbreviations	R U K C RU 2 4 BRB LOL BTW JK L8R LMAO NP IDK OMG TTYL TTYS WTF FYI ASAP IC CU OIC PLS PLZ CYA ROTFL THX IDC OTP U2 YT IMHO …
Sentence Structure	Average words per sentence

words such as ru (are you) and 4 (for). Table 1 shows the stylometric features that may be collected and analyzed for IM author classification.

5. Data Description

The experiments used IM conversation logs collected by the Gaim and Adium clients. The IM conversations were logged to ASCII text files in the following format:

[timestamp] [user name:] [message]

For example:

(14:19:29) User1: hey, what time is the meeting today?

(14:19:35) User2: It is at 11AM…are you going?

(14:19:39) User1: yeah, I will be there, it sounds very interesting! :) :)

The data required a series of preprocessing steps. First, the raw IM logs were parsed to extract data for each user. The data were prepared for analysis by removing all entries that did not belong to the user under analysis (UserX), as well as removing both the timestamp and username. Thus, an example of a formatted log for User 1 looks like the following:

hey, what time is the meeting today?

yeah, I will be there, it sounds very interesting! :) :)

Next, the logs were split into 2500 character segments to create instances. Sometimes this was a complete single conversation log, and sometimes it involved combining smaller conversation logs to meet the required length. Finally, the instances were processed to generate frequency totals for each attribute of a behavior category for the author. The frequency total data was outputted in CSV format and formatted as a Weka data file.

The data used in this research consisted of IM conversation logs for four users categorized in the following classes: User1, User2, User3, and User4. The data was parsed to calculate the sentence structure and the frequencies of predefined special characters, emoticons, and abbreviations, resulting in a total of 69 numeric at-

Table 3
IM Data Classification Results

J48	Overall Accuracy: **97.86%** Error: **2.14%**		
	TP	**FP**	
User1	.97	.01	
User2	1	.019	
User3	.97	0	
User4	.97	0	
IBk	Overall Accuracy: **97.14%** Error: **2.86%**		
	TP	**FP**	
User1	.97	0	
User2	.97	.029	
User3	.94	.01	
User4	1	0	
Naïve Bayes	Overall Accuracy: **99.29%** Error: **0.71%**		
	TP	**FP**	
User1	.1	.01	
User2	1	0	
User3	1	0	
User4	.97	0	

```
   a    b    c    d     <--
classified as
 34    1    0    0  |   a = User1
  0   35    0    0  |   b = User2
  0    1   34    0  |   c = User3
  1    0    0   34  |   d = User4
```

```
   a    b    c    d     <--
classified as
 34    1    0    0  |   a = User1
  0   34    1    0  |   b = User2
  0    2   33    0  |   c = User3
  0    0    0   35  |   d = User4
```

```
   a    b    c    d     <--
classified as
 35    0    0    0  |   a = User1
  0   35    0    0  |   b = User2
  0    0   35    0  |   c = User3
  1    0    0   34  |   d = User4
```

Table 4
Classification Accuracy Results for Attribute Categories

Classification Method	Special Characters	Emoticons	Abbreviations
J48	91.43%	55.71%	95.71%
IBk	92.86%	50%	89.29%
Naïve Bayes	92.14%	56.42%	97.85%

tributes. Table 2 shows the 1 sentence structure attribute, 17 special characters, 16 emoticons, and 35 abbreviations defined in this study.

The conversation log data was parsed to create 35 instances for each class, for a total of 140 instances. The original data was unbalanced, since some users had more frequent and longer conversations than others. Therefore, the data was undersampled to the size of the smallest dataset to create a balanced dataset, resulting in 35 instances of 2500 characters per class.

6. Classification Methods and Results

The experiments used various classification methods to determine if an author of an IM conversation could be identified based on his or her sentence structure and use of special characters, emoticons, and abbreviations. The experiments also determined which features were strongest at identifying authors. The research used the Weka data mining tool for classification.

The classification methods used for this research were the J48 decision tree, IBk nearest neighbor, and Naïve Bayes classifiers. Table 3 shows the accuracy, error, true positive (TP), and false positive (FP) rate of each classifier when applied to the dataset.

Next, individual attribute categories were tested. Table 4 shows the accuracy results for each classifier when applied to each individual attribute category.

Attribute selection was used to rank the strongest identifying attributes. Table 5 shows the strongest identifying attributes according to the information gain and chi-squared techniques.

Table 5
Attribute Selection

Information Gain	Chi-squared
U	U
…	…
-	-
.	.
,	‘

Next, the top 3 individual attributes (U, three dots, and the hyphen) were tested individually with each classifier. Table 6 shows the accuracy results for each classifier when applied to each individual attribute.

Table 6
Classification Results for Top Attributes

Classi-fication Method	U	…	-
J48	62.86%	67.86%	61.43%
IBk	66.43%	67.86%	61.43%
Naïve Bayes	65.71%	68.57%	62.86%

Experiments using classification methods on the IM datasets resulted in the following conclusions:

- Abbreviations were the best discriminators with 97.85% accuracy, followed closely by special characters with 92.86% accuracy.

- The Naïve Bayes classification method performed with only the abbreviations attributes (97.85%) resulted in a higher accuracy than the J48 classifier (97.86%) and with a similar accuracy as the IBk classifier (97.14%) when all attributes combined were used in J48 and IBk.
- The strongest identifying attributes were U, three dots, and the hypen.
- None of the individual attributes identified by attribute selection were strong enough to determine author identification with a high degree of accuracy.
- The combination of all attribute categories using the Naïve Bayes classification method provided the best results (99.29% accuracy).

7. Summary and Future Work

The IM market has seen explosive growth, with millions of users participating in online conversations. However, little has been explored in terms of the research and analysis of the network, messages, user behavior, and data mining of these systems. There are several concerns involving the use of IM systems, including whether the user is really communicating with the intended buddy or friend. The threats include account hijacking, man-in-the-middle attacks, and masquerading. There are various reasons someone would wish to masquerade as someone else including spying, disgruntlement, snooping, and other malicious intentions.

This paper presented the use of data mining of IM communications for authorship identification. Classification methods were used to identify IM authors based on various behaviors. Human behavior presents challenges for analysis. For example, such behavior has an extremely wide "normal" range and can be very unpredictable: abnormal activities are sometimes perfectly normal, and all people change. The results of the experiments here indicate that Naïve Bayes classification is highly accurate (> 99% accuracy) at predicting the author of an IM conversation based on behavior. The experiments also identified the behavior characteristics that are the strongest classifiers. The data showed that users tend to exhibit the same characteristics throughout various conversations. Furthermore, users exhibited different characteristics from each other, much like the uniqueness of biometric data.

Based on these preliminary experiments, future research will involve:

- Increased numbers of users (classes) in the dataset.
- Increased numbers of attributes from the set of stylometric features, such as characters, function words, and structural layouts.
- Varied numbers of characters that are included in an instance, to determine the minimum size necessary for high accuracy and a low false-positive rate.

The author behavior attributes used in these experiments comprise only a subset of the stylometric features that may be used for IM author identification. Other stylometric measures, as shown in Table 1, must also be used to create an accurate, well-rounded user profile. A broad attribute set and larger number of classes should provide a comprehensive analysis of the IM data for highly accurate authorship identification.

8. References

[1] Corney, Malcolm. Personal Website. http://sky.fit.qut.edu.au/~corneym.

[2] Donaldson, Tom. "Computational Analysis of Verbal Behavior". http://behavioranalysis.mactom.com/cavb.html.

[3] Goldring, Tom. "Authenticating Users by Profiling Behavior". http://www.cs.fit.edu/~pkc/dmsec03/slides/goldring03dmsec.ppt.

[4] O. de Vel, A. Anderson, M. Corney and G. Mohay, "Email Authorship Attribution for Computer Forensics", in Daniel Barbara, Sushil Jajodia, "Applications of Data Mining in Computer Security", ISBN 1-4020-7054-3, Kluwer Academic Publishers, Boston, 2002, 252 pages.

[5] O. de Vel, A. Anderson, M. Corney and G. Mohay. "Gender-Preferential Text Mining of E-mail Discourse".18th Annual Computer Security Applications Conference (2002 ACSAC) December 9 – 14, 2002, Las Vegas, Nevada, USA.

[6] O. de Vel, A. Anderson, M. Corney and G. Mohay. "Language and Gender Author Cohort Analysis of E-mail for Computer Forensics". Digital Forensic Research Workshop, August 7 – 9, 2002, Syracuse, NY, USA.

[7] O. de Vel, A. Anderson, M. Corney & G. Mohay, "Mining Email Content for Author Identification Forensics", SIGMOD Record Web Edition, 2001, 30(4).

[8] O. de Vel, A. Anderson, M. Corney & G. Mohay, "Multi-Topic E-mail Authorship Attribution Forensics", ACM Conference on Computer Security - Workshop on Data Mining for Security Applications, November 8, 2001, Philadelphia, PA, USA.

[9] Resig, John. Ankur Teredesai. "A Framework for Mining Instant Messaging Services". SIAM DM 2004 Workshop on Link Analysis, Counter-Terrorism & Privacy.

[10] Resig, John. Santosh Dawara, Christopher M. Homan, and Ankur Teredesai. "Extracting Social Networks from Instant Messaging Populations". KDD 04 Link Discovery Workshop (LinkKDD 2004).

[11] Rong Zheng, Yi Qin, Zan Huang, Hsinchun Chen. "Applying Authorship Analysis in Cybercrime Investigation". Lecture Notes in Computer Science. Publisher: Springer-Verlag GmbH. ISSN: 0302-9743. Volume 2665 / 2003. Chapter: pp. 59 – 73.

[12] Szymanski, Boleslaw. "Recursive Data Mining for Masquerade Detection and Author Identification". 5th Annual IEEE Information Assurance Workshop. June 2004. http://www.cs.rpi.edu/~szymansk/papers/ia04.pdf.

[13] Zhou, Lina, Zhang Dongsong. "Can Online Behavior Unveil Deceivers?", Proceedings of the 37th Hawaii International Conference on System Sciences. 2004.

Angela Orebaugh is a cyber security technologist leading a variety of security innovation projects including research for the National Institute of Standards and Technology. She has 15 years experience in information technology and security and is the author of several technical security books. Ms. Orebaugh is an adjunct professor for George Mason University where she is completing her PhD with a focus on digital forensics and cybercrime.

Dr. Jeremy Allnutt is a Professor in the Department of Electrical and Computer Engineering as well as the Director of the Masters in Telecommunications Program at George Mason University. He has recently created the new Masters in Computer Forensics program at GMU that prepares students for careers in industry, government, and academia by combining academic education with real-world practical techniques.

IJoFCS (2009) 1, 29-42

e-forensic PRESS

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

Autonomic Forensics a New Frontier to Computer Crime Investigation Management

José Helano Matos Nogueira(1), and Joaquim Celestino Júnior(2)

(1)Technical-Scientific Sector, Brazilian Federal Police, Fortaleza – Brazil, helano@apcf.org.br
(2) LARCES/UECE, State University of Ceará, Fortaleza – Brazil, celestino@larces.uece.br

Abstract - The remarkable growth of investigations and exams numbers accomplished in computer crime investigation, as well as the integration of a variety of different technologies with the goal of providing quality of services, has transformed computer forensic management into an extremely complex activity. As the complexity continues to increase, it will be necessary to delegate management tasks to the machines themselves. The forensic systems and tools will need to execute activities that are currently performed by experts and investigators, in a fast and transparent way, with few or no mistakes. These will be the autonomic forensics: a new paradigm that defines the method of management. Hence, the main challenge of this work is to use autonomic computing to create this paradigm and apply it to computer forensic management.

Index Terms - Autonomic computing, computer crime investigation, forensic computer, management.

1. Introduction

Some years ago, IBM released a visionary manifesto stating that the main obstacle to progress in Information Technology (IT) was a looming software complexity crisis [1]. Although this document was visionary for its time, it is now cited by many authors, researchers, and technological centers [2, 3, 4]. As discussed in the manifesto, computational systems have evolved considerably since their inception, making the management of these systems extremely complex. Several reasons contribute to the need for complex management of the systems, including: 1) the increased need for interconnectivity, 2) integration of heterogeneous hardware, 3) development of novel software and technology, and 4) difficulty in the satisfactory allocation of computational resources [5]. The same need for complex management also occurs with computer crime investigations. Criminal experts and investigators are required to collect collect, preserve, and save pieces of evidence. Many different software programs are necessary to rescue, search, relate, filter, monitor, and analyze the many pieces of evidence. Besides, it is fundamental to pursue procedures, techniques, methods that these police persecutors need to follow at each phase of the forensics.

Many tasks and processes associated with today's forensic computer infrastructure are rapidly becoming outdated, obsolete or outmoded.

Gathering 10 or 20 computer experts to pore over large hard disc files, sniffer traces, and log files to resolve a computer investigation is prohibitively expensive and too time consuming to address today's need for near perfect service uptime. Once the methodology of crime scene investigation is changed, administrators and experts in the future years will look back on the current technology and say, "When did we have time to actually investigate the incident?" To cope with the ever increasing complexity of management, we demonstrate how autonomic computing can reduce the burden on investigators overwhelmed by the current technology.

This paper is organized as follows. In section II, we briefly outline the autonomic computing. We describe the roadmap of conceptualization, development of autonomic computing, and the requirements essential to the creation of an autonomic forensic system. In section III, we define a realistic ontology for computer crime investigation with units, resources (hardware, software), hierarchies, classes, and objects, including the relationships among these elements. Finally, in section IV, we present the main goal this work when we describe in detail, our proposal for autonomic forensic management architecture. In this new architecture, we present the structural design, management layers, proposal architecture, and open standards for implementation.

2. Autonomic Computing

2.1. Roadmap Conceptualization

The inspiration for the term "autonomic computing" is derived from the autonomic nervous system (ANS), shown in Figure 1. The human nervous system is divided into the voluntary and involuntary systems. The ANS is the involuntary portion of the nervous system and controls the heartbeat, digestion, circulation, and glandular functions. The ANS can be further divided into subsystems, including the sympathetic nervous system and the parasympathetic nervous system [6].

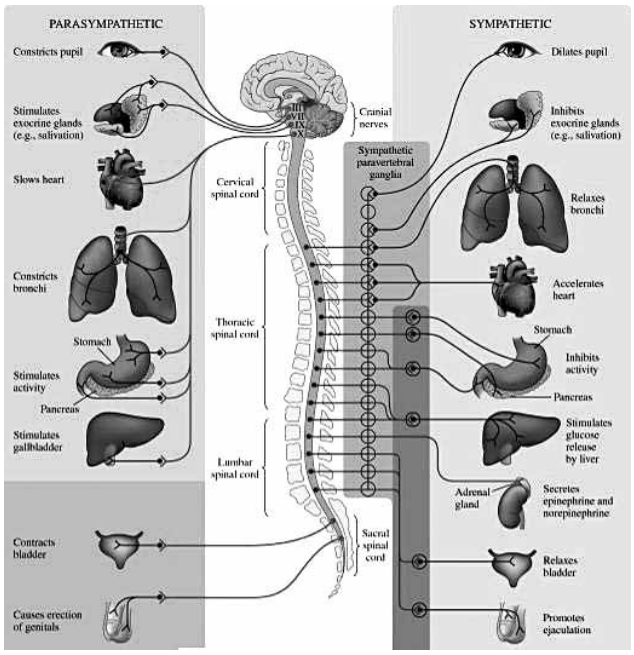


Fig. 1. The human ANS and some of the features automatically regulated

We take for granted the human body's ability to self manage all of its organs and systems. Similarly, we take for granted the computer's ability for self-management. Hence, the essence of autonomic computing systems is self-management [7, 8, 9]. Autonomic computing intends to free system administrators from the details of system operation and maintenance and to provide users with a machine that runs at peak performance 24 hours per day 7 days a week. In the vision of Kephart and Chess [10], "the autonomic computing has new components integrate as effortlessly as a new cell establishes itself in the human body. These ideas are not science fiction, but elements of the grand challenge to create self-managing computing systems."

Is there complexity in computer forensics?

First, we need to define what complexity is in our approach. In this instance, complexity is the proliferation of heterogeneous resources (hardware and software), as well as technology environments, which require the components of a given solution to be integrated and customized into a unique strategic process. Here are some of the symptoms of complexity in an IT infrastructure as it applies to forensics (**IT-Forensics**):

1. Frequent and recurring software crashes of critical forensic applications due to incompatibility of data, files, errors, or network protocols.
2. Longer timeframes for forensic staff to solve the problems listed in item 1.
3. A significant increase in IT-Forensic budgets, including hardware, software, human capital costs, training, and support.
4. Increase in the level of application outsourcing.
5. High turnover of critical forensic staff due to frustration, long hours, and burnout.
6. Unexpected surprises in new technology, new languages, or applications leading to increased time in understanding and managing projects that use them.
7. Longer timeframes to satisfactorily test and install new applications or software packages.
8. Growth of expensive hardware and software in IT-Forensics.
9. Incompatibility between competing supplier software packages—i.e., file structures, databases, and parameters—due to lack of standards.
10. Frequent, but necessary software upgrades of packages, forensic tools, and operating systems, resulting in another round of errors and problems with incompatibility.

Why use autonomic computing to forensics?

The answer to this question is simple and feasible. The remarkable growth of investigations and exams numbers accomplished in computer crime investigation, as well as the integration of a variety of different technologies with the goal of providing quality of services, has transformed the management of computer forensics systems into a very complex activity. With the marked increase in complexity, it will be necessary to delegate management tasks to the machines themselves. Hence, autonomic computing is the solution to decrease this complexity via the creation of self-management mechanisms. We propose to use the autonomic environment to force the forensic systems and tools to execute activities that are currently performed by experts and investigators, in a fast and transparent way, with few or no mistakes.

What if the unthinkable happens, and we do not adopt autonomic computing or similar technology within a few years?

If this technology is not incorporated, some of the following events will happen:

1. Complexity will continue to increase, reaching proportions that are not manageable.
2. The pressure on forensic staff (mainly with the computer forensic experts) to fix unfixable problems will increase.
3. Reliability of services, systems and performance will deteriorate and computer forensics as a whole will suffer.
4. Forensic corporations (police forces and forensic organizations) will demonstrate a decline in performance levels and will lose substantial time performing appropriate duties.
5. Forensic corporations will be required to increase their IT-Forensics budgets.
6. High staff management and their directors will reject requests for budget increases and the cycle of problems will continue.
7. Additional skilled forensic staff will be needed at substantial costs.
8. The health of many forensic staff will suffer.
9. Chaos will be established.

1.2. Developing Autonomic Computing

Autonomic computing and the subsequent self-management is an evolution, not a revolution. Delivering system-wide autonomic environments is

Table 1
The path to autonom

Basic (Level 1)	Managed (Level 2)	Predictive (Level 3)	Adaptive (Level 4)	Autonomic (Level 5)
Features Reactive. Multiple sources of system generate data. Manual analysis and problem solving.	Features Consolidation of data and actions through management tools.	Features Proactive. System monitors, correlates and recommends actions.	Features System monitors, correlates and recommends actions.	Features Integrated components dynamically managed according to strategic rules/policies.
Tools Local, platform and product specific.	Tools Consolidated resource management consoles, problem management system, automated software install, intrusion detection, load balancing.	Tools Role-based consoles with analysis and recommendations; product configuration advisors; real-time view of current and future IT-Forensics performance; automation of some repetitive tasks; common knowledge base of inventory and dependency management.	Tools Policy management tools drive dynamic change based on resource specific policies.	Tools Costing/financial analysis tools, business and IT-Forensic modeling tools, tradeoff analysis; automation of some e-business management roles.
Skills Require extensive, highly skilled forensic staff .	Skills Forensic staff analyzes and takes actions. Multiple management tool skills.	Skills Forensic staff approves and initiates actions.	Skills Forensic staff manages performance against service level agreements.	Skills Forensic staff focuses on enabling business needs.
Benefits Time to fix problems and finish tasks.	Benefits Greater system awareness. Improved productivity.	Benefits Reduced dependency on deep skills. Faster/better decision making.	Benefits Balanced human/ system interaction. IT-Forensic agility and resiliency.	Benefits Strategic policy drives Forensic management. Business agility and resiliency.

an evolutionary process enabled by technology. However autonomic computing must ultimately be implemented by each organization through the integration of these technologies and supporting processes into current computational paradigms. The path to autonomic computing can be thought of in five levels: basic, managed, predictive, adaptive, and autonomic [11]. To apply these levels to the forensic area, we have adapted the levels as presented in Table 1.

The basic level represents the starting point for many forensic organizations. If forensic organizations are formally measured, they are typically evaluated on the time required to finish major tasks and fix major problems.

In the managed level, forensic organizations are measured on the availability of their managed resources, their time to close trouble tickets in their problem management system and their time to complete formally tracked work requests. Forensic organizations improve efficiency through the consolidation of management tools to a set of strategic platforms and through a hierarchical problem management triage organization.

In the predictive level, forensic organizations are measured on the availability and performance of their strategic forensic systems and their return on computer crime solution. The critical nature of the forensic organization's role in the success of the investigation is understood. Predictive tools are used to project forensic performance and recommendations based on these projections are made to improve future performance.

In the adaptive level, IT-Forensics resources are automatically provisioned and tuned to optimize transaction performance. Investigative policies, investigative priorities, and service-level agreements guide the autonomic infrastructure behavior. Forensic organizations are measured on response times (transaction performance), the degree of efficiency of the IT-Forensics and their ability to adapt to shifting workloads.

In the autonomic level, forensic organizations are measured on their ability to make the computer crime investigation successful. To improve strategic investigation measurements they understand the metrics associated with Criminalistics activities and supporting IT-Forensics capabilities. Advanced modeling techniques, including artificial intelligence, are used to optimize investigation performance and quickly deploy newly optimized solutions to computer crimes.

1.3. Requirements to Build Autonomic Forensic Systems

To be truly self-managed, a computing system needs to "know and understand itself," while being comprised of components that also possess a system identity [2]. Since a "paradigm" can exist at many levels, an autonomic forensic system will need detailed knowledge of its components, investigative capacities, current status, operating environment, and of all connections with other systems. As shown in Figure 2, an autonomic system with ability to self-manage must possess four basic characteristics [1, 10]: self-configuring, self-optimizing, self-healing, and self-protecting.

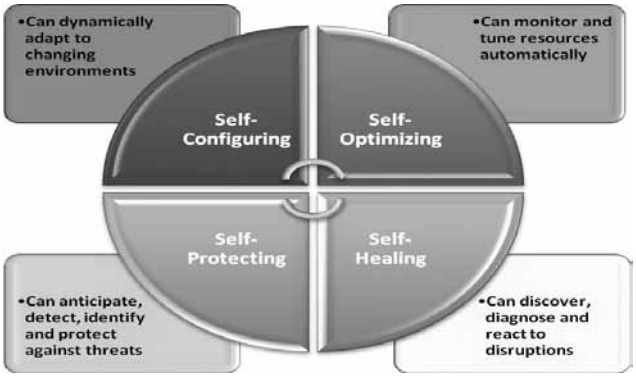


Fig. 2. The four basic features of an autonomic forensic management system

Self-configuring

Autonomic forensic management systems need the ability to dynamically adapt to changing environments. Installing, configuring, and integrating complex systems are challenging, time-consuming, and error-prone even for forensic experts. An autonomic system must be able to install and set up software automatically in accordance with high-level policies, which represent forensic goals.

Examples:

1. Installation, testing, and release of regular supplier service packs.
2. Installation of supplier patches, corrections, and modifications together with the necessary testing and release.
3. Automatic and seamless installation of new forensic software.

Self-optimizing

Autonomic forensic management systems need to automatically monitor and tune resources. Complex software, such as Forensic ToolKit-FTK® or Encase®, or database systems, such as Oracle® or MySQL®, have hundreds of adjustable parameters that must be correctly set for optimal system performance, however few people know how to adjust them. Such systems are often integrated with other, equally complex systems. Consequently, performance-tuning of one large subsystem can have unanticipated effects on the entire system.

An autonomic system must constantly monitor predefined system goals or performance levels to ensure that all systems are running at optimum levels. With the strategic policies constantly changing and demands from forensic customers and suppliers changing equally fast, self-adapting requirements are needed.

Examples:

1. Optimum sub-second response times for all types of access devices, such as personal computers, notebooks, PDAs, duplication devices, and media peripherals.
2. Interfacing with other modules to exchange data and files.
3. Working with forensic software from outside suppliers.

Self-healing

Autonomic forensic management systems need to discover, diagnose and react to disruptions. The system must also automatically detect, diagnose, and repair localized software and hardware problems. Autonomic systems will have the ability to discover and repair potential problems to ensure that the systems run smoothly. Self-healing systems will be able to take immediate action to resolve the issue, even if further analysis is required. Rules for self-healing will need to be defined and applied. As autonomic systems become more sophisticated, embedded intelligence will be applied to discover new rules and objectives.

Examples:

1. When a process fails, the errors or problems are identified and processes are re-run without human intervention.
2. When a database index fails in a forensic tool the files are automatically re-indexed, tested, and loaded back into production.
3. File space and database storage is automatically extended according to previous data on growth and expansion.

A good example of self-healing can be found in the SMART (Self-Managing and Resource Tuning) database from IBM [2]. This database is designed to run with minimal need for human intervention. In SMART the user can opt not to be involved, and the database will automatically detect failures as they occur (and correct them) and will configure itself by installing operating systems and data automatically to cope with the changing demands of e-business and the Internet.

Self-protecting

Autonomic forensic management systems need to anticipate, detect, identify, and protect against threats. Despite the existence of firewalls and intrusion detection tools, currently humans must decide how to protect systems from malicious attacks and inadvertent cascading failures. Autonomic system solutions must address all aspects of system security at the platform, operating system, network, and forensic application. Self-protecting components can detect security incidents as they occur and take corrective actions to make themselves less vulnerable.

To achieve, continuous sensors that feed data to a protection center are required. A log of events will need to be written and accessed when appropriate for audit purposes. To manage the threat levels, a tiered level might be expected. Threats can be escalated through the tiers for increasing action and priority.

Examples:

1. Implement tiered security levels.
2. Target resources on network monitoring and immediately disconnect computer systems with suspicious network traffic.
3. Verify that network configurations and security policies are correct and, if not, take action.

1.4. Ontology - Theory and Practice

Ontology is an explicit specification of a conceptualization of the real world [12]. In such an ontology, definitions associate the names of entities in a universe (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and providing formal axioms that constrain the interpretation and proper use of these terms. Hence, an ontology defines a formal common vocabulary for researchers who need to share information in an application domain.

Why would someone want
to develop an ontology?

Before beginning any elaboration of a forensic autonomic architecture, it is necessary to understand the scope of computer crime investigation. Therefore, we created the Autonomic Forensic Ontology (AFO) to map the forensic application domain (Figure 3). Among the

advantages that have been presented in literature for the adoption of ontologies [13, 14, 15], the following important advantages are highlighted:

- ✓ Sharing: the use of ontology permits a common understanding about a knowledge domain;
- ✓ Reuse: the use of explicit and formal definitions simplify knowledge maintenance, creating agreement between users for a given domain model and, thus facilitating the ontology reutilization;
- ✓ Information Structuring: allows for the capture of data semantics and automatic processing, while the terminology is understandable to the human user;
- ✓ Interoperability: it allows information sharing among different computational systems;
- ✓ Reliability: an ontology-based information representation makes possible a consistent and more trustworthy implementation;
- ✓ Distinction: separates domain knowledge from the operational knowledge.

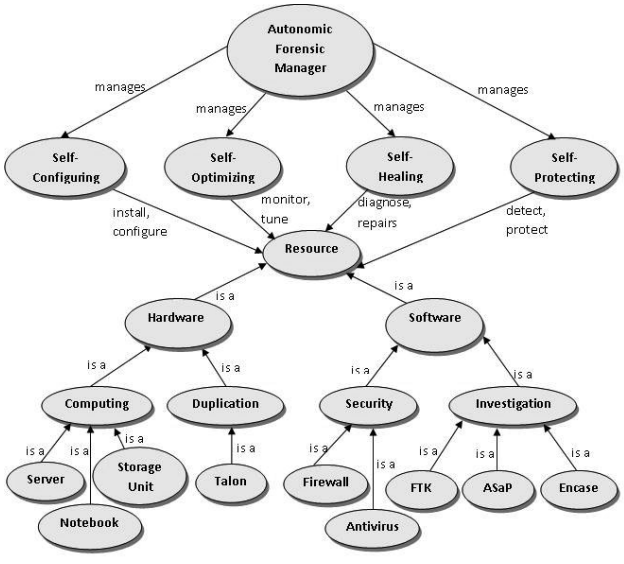


Fig. 3. Part of AFO for computer crime investigation

In AFO, we created an autonomic representation the four basic characteristics of self-management: self-configuring, self-optimizing, self-healing, and self-protecting. The ellipses describe domain concepts while the arrows among the concepts are the relationships. Relations labeled "is-a", describe the existence of specialization relations (subclasses), which create hierarchies among the concepts. In this way, we intend to go in the same direction of author [13].

After that, it is important to develop an ontology solution using a particular software tool. To achieve this purpose, we investigated several ontology computational languages: Chimaera [16], Ontolingua [17], and Protégé [18, 19]. Pilot testing revealed that Prótégé was more appropriate for forensic applications (Figure 4).

AFO supplies the basis for a formal and explicit knowledge to support computer crime investigation management in autonomic systems. The main advantage of this approach is that it enables the use of sophisticated processes of reasoning and, consequently, the follows the established efficient decision making methods employed by the computer experts or forensic investigators.

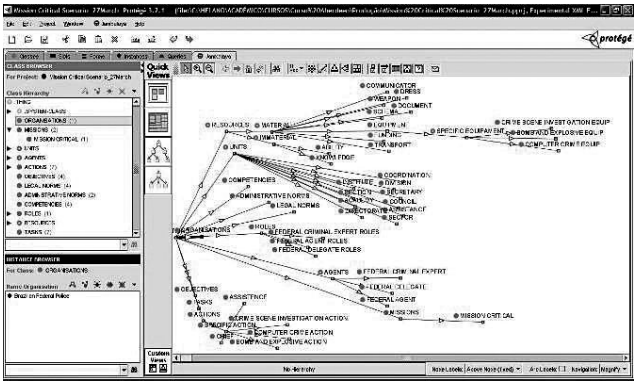


Fig. 4. AFO ontology implemented using Protégé

3. Method for Autonomic Forensic Management

3.1. Structural Design

Traditionally, IT-Forensics infrastructures have been organized on individual categories, separated by both component type and platform type. For example, an investigator

might be concerned with managing only one portion of the infrastructure. This may be security tools, internet investigation programs, databases, or stand-alone forensic applications. Hence, the creation of Autonomic Forensic Management Architecture (AFoMA) formalized a reference framework that identified common functions across all categories. AFoMA set forth the building modules required to achieve autonomic managing.

The concepts presented in AFoMA define a common approach and terminology for describing autonomic forensic paradigms. Thus, AFoMA organizes an autonomic management into the layers shown in figures 5 and 6. These layers are connected using the Multi-Agent Systems (MAS) that allows the components to collaborate and to have code mobility [20, 21, 22]. In general, MAS are considered real or virtual entities that emerge in an environment in which they 1) can make decisions, 2) are capable of noticing and representing, at least partially, this environment and 3) are capable of communicating with other agents autonomously [23, 24]. These characteristics are consequences of an agent's observation, knowledge and interaction with other agents. Each agent has an internal decision-making system that acts globally, around surrounding agents, but is also capable of working alone. In this way, an agent possesses perception mechanisms to comprehend its environment [25, 26, 27]. Although there still is not a consensus on a formal definition of an agent in this context, some expected characteristics have been established.

We chose MAS to obtain the autonomic properties because these systems possess natural and desirable characteristics including:

- ✓ Mobility: The agent's ability to move among layers;
- ✓ Helpful: the agent does not have contradictory objectives, and the agent will always attempt to perform the requested task;
- ✓ Rationality: the agent will act to achieve its objectives;
- ✓ Adaptability: an agent has the ability to adjust to the habits, work methods and preferences of its manager or user;
- ✓ Collaboration: an agent should not accept and execute instructions without considerations, but it should take into account that the human user commits mistakes, omits important information and supplies ambiguous information. In this case, an intelligent agent should check these occurrences and ask questions of the user.

Hence, at each inter-layer in AFoMA, one or several agents are downloaded. Additionally, the deployed agents may be identical or different from one layer to another. More precisely, they differ according to the objectives assigned to them. However, all the agents keep the same internal architecture.

The main functionalities of the MAS model under the AFoMA architecture are:

- ✓ To communicate inside the architecture (inter-agent communications for mutual information exchange);
- ✓ To collect all the information (knowledge) from the local layer and from the neighborhood;
- ✓ To collect useful information from the autonomic element, such as: traffic, conflicts, application services, routing information;
- ✓ To perform actions and changes on the architecture layers;
- ✓ To pilot (govern) the behavior of each agent.

2.1. Management Layers

We propose a management model divided in five layers or modules, shown in figure 5.

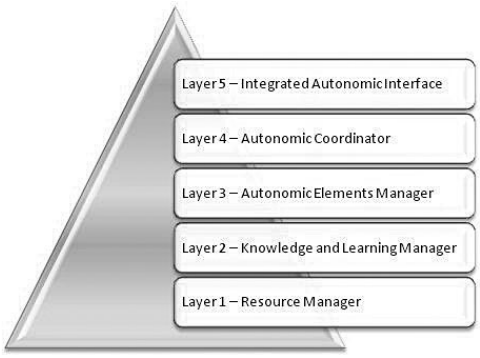


Fig. 5. Management Layers in AFoMA

Each layer has functionalities and specific services. In the top of the pyramid (layer 5) the integrated autonomic interface is addressed. This interface is the unique contact point between the user and the autonomic architecture and is the place where strategies and politics are defined. In the base (layer 1), the operational manager that manages the infrastructure resources is established. Several elements are addressed by the middle of the pyramid (layers 2, 3 and 4), particularly the knowledge and learning manager that controls all the knowledge repositories and the deduction module; the autonomic elements manager that individually manages each autonomic component (self-configuring, self-optimizing, self-healing, and self-protecting); and the autonomic coordinator that collectively harmonizes the autonomic components.

All layers that can achieve management functions without human intervention (layers 1 to 4) execute a continuous cycle called ICL (Intelligent Control Loop), consisting of operations that include monitoring, analysis, planning and execution [10, 11]. More functional details of each layer are given in the next section where the autonomic forensic management architecture is defined.

2.2. Integrated Architecture

The global managing model has been presented in the previous section. Such a model can be generally applied to various applications. However, at this point, it is necessary to explain how to use the autonomic paradigm for realistic forensic scenarios. This is where the proposed Autonomic Forensic Management Architecture (AFoMA) becomes relevant. It is graphically described in figure 6. It is important to note that AFoMA is divided into the same management layers as previously described.

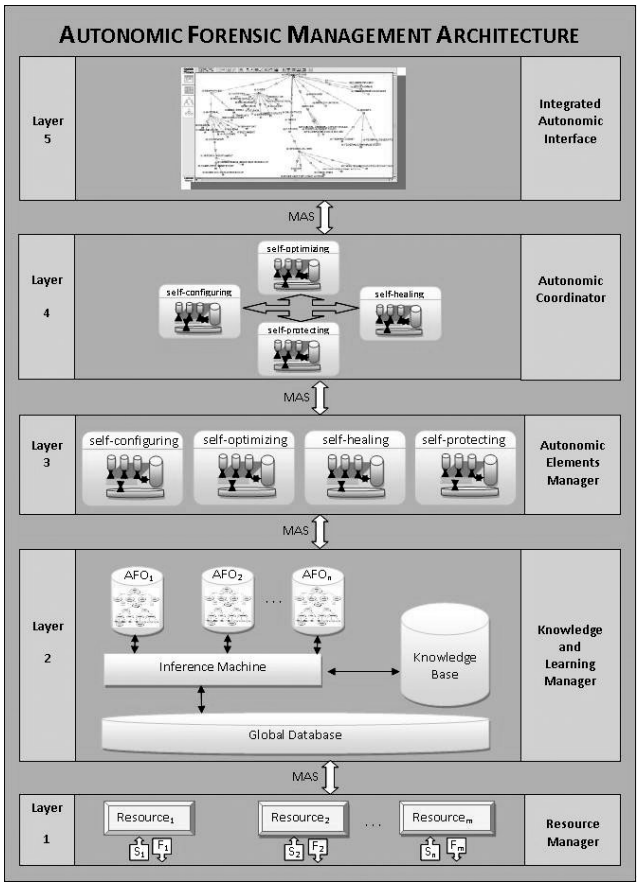


Fig. 6. AFoMA Architecture

Resource Manager

This first layer contains the Resource Manager that manages the system components, or resources, that make up the IT-Forensic infrastructure. All managed resources need to be mapped in AFO ontology (Figure 3) because this element represents the application domain. A resource is a hardware or software component that can be managed. Resources can refer to servers, storage units, imaging devices, security software, investigation tools, services, applications or other entities.

As shown in Figure 6, each resource has a sensor (S_i), $1 \leq i \leq m$, and an effector (F_i), $1 \leq$

$i \leq m$. Sensors and effectors allow the manager to receive valuable information and to perform actions upon the managed resource, respectively. A sensor is a device or interface made to receive information by something outside the resource. A sensor monitors the resource receiving information about the state and state transitions of this managed resource. For example, sensors get information about the current state of a managed resource or retrieve asynchronous messages or notifications. An effector is a device or interface made to react by something outside the changing the resource state. An effector acts in agreement with the results of the sensor. For example, effectors can request operations to allow the resource to consult with some external entity.

In IBM architecture [1] the sensors and effectors of the autonomic manager are used to transmit and receive data and associated information. In AFoMA, we propose one important difference: the sensors and effectors are projected to be controlled by rules, norms or contracts [28, 29], which in addition to the previous benefits, improves the level of knowledge (e.g., differentiating between data, facts, beliefs, rules, norms, and contracts). This defines a taxonomy, and enables semantics to be defined using the AFO ontology approach described earlier.

Knowledge and Learning Manager

This second layer contains the Knowledge and Learning Manager that can be divided into two parts: the repositories and the inference machine. The repositories, also called bases, store the ontology knowledge bases and the global databases. The ontologies, as described in section III, store the knowledge over the application domain with their entities (classes, objects, instances) and their relationships. In our forensic approach the ontologies possess all AFOs with their autonomic functionalities. In our proposal, the knowledge base is formed by facts, beliefs, rules, norms and contracts. It is therefore possible to obtain a deeper knowledge of an application domain. In this knowledge base, part of the experiences, learning, and knowledge are stored. The global database store data, information and parameters received by the sensor and effectors, as well as the temporary data obtained from transitions, logs and control variables. The inference machine is the module of deductions of the proposed architecture. The inference machine accomplishes deductions, inductions, and data-mining from the elements stored in the repositories, described previously. Here the facts and beliefs can be used as absolute truth values (tautologies). However, this machine should be capable of treating more elaborated elements as rules and norms that formed the knowledge and the learning in AFoMA. As shown in figure 6, the inference machine makes the connections among the repositories. Additionally, the inference machine should be capable of using all previous elements in future contracts [30, 31]. Contracts here can be defined by rules and norms that define the politics or strategic goals [32, 33]. So, contracts are used to guide for self-management in our autonomic model. Two additional autonomic characteristics of this inference machine are to create the self-knowledge once storage of the data, knowledge and experiences is possible and to make deductions that generate the self-learning.

Autonomic Elements Manager

This third layer contains the Autonomic Elements Manager. It manages each autonomic element (self-configuration, self-optimization, self-healing, and self-protection) in an individualized way. Autonomic elements have an internal Intelligent Control Loop (ICL, section IV–B) to monitor, analyze, plan, and execute their activities and tasks. So, an element in this layer is seen in an isolated fashion and each element is responsible for goals and tasks already described in section II-C. Some examples of such tasks, using the four self-managing categories introduced earlier in this paper, include:

- ✓ Performing a self-configuring task such as installing software when it detects that some prerequisite software Performing a self-optimizing task such as

adjusting the current capacity when it observes an increase or decrease in workload;

- ✓ Performing a self-healing task such as correcting a configured path so installed software can be correctly located;
- ✓ Performing a self-protecting task such as taking resources offline if it detects an intrusion attempt.

Also, it is in this layer that the new self-management characteristics are increased. Thus, a system or architecture can begin with one or few autonomic elements and develop new autonomic elements in this layer.

Autonomic Coordinator

This fourth layer contains the Autonomic Coordinator. A single autonomic element acting in isolation can achieve autonomic behavior only for the resources that it manages. The self-managing capabilities delivered by Autonomic Elements Manager need to be coordinated to deliver architecture-wide autonomic computing behavior. The autonomic coordinator provides this coordination function. It coordinates the communication protocols among elements described in the previous layer. Such protocols can be verified when it is necessary to treat priorities, exceptions or to solve conflicts among the autonomic elements that actuate in the layer below. The autonomic coordinator also implements the ICL (section IV-B) that automate combinations of the tasks found in one or several autonomic elements.

Integrated Autonomic Interface

This fifth layer contains the Integrated Autonomic Interface. It is composed by the user interface for the autonomic managers (layers 1 to 3) and the coordinator (layer 4). In some cases, an administrator might choose that certain tasks will involve human intervention, and the human interaction with the system can be enhanced using the console of the Integrated Autonomic Interface. It is here that the user has contact with the proposed architecture of self-management - AFoMA. The use of an integrated interface reduces the cost of ownership (attributable to more efficient administration) and creates a familiar user interface, reducing the need for staff to learn a different interface each time a new product is introduced. Thus, an integrated autonomic interface consists of a common interface and specific components provided by AFoMA.

2.3. Open Standards to Implementation

We have already identified that the IT organizations are going through major changes. New technology, such as autonomic computing, web services, and grid computing are creating tremendous opportunities to massively increase forensic profitability. The potential of these technologies to transform computer crime investigation is amazing, and open standards will play a critical role in this new on-demand world.

The open standards community includes major companies such as IBM, HP, Sun, Motorola, Intel, and Microsoft that are active contributors to autonomic computing [35]. These examples are only a small portion of the hundreds of other companies involved in these initiatives. Contributors from the academic world, including as Stanford, Berkeley, and MIT, also account for a large portion of the open standards community.

> Why are major IT infrastructure and software companies increasingly interested in open standards to autonomic computing?

The global computer industry must cooperate in developing the necessary open standards and interfaces to make future technology work and to establish standards that will support an autonomic environment. Therefore, open standards are essential to the success of autonomic computing. Many organizations strongly agree that open standards are imperative for autonomic environments (Figure 7).

The adoption of open standards may appear to be a daunting task, as each component of

autonomic computing will need to "describe" itself to other software, their resources, and most importantly, their requirements. For example, the self-configuration component will need to contact a supplier when it detects that some prerequisite software is missing and state: "Send me the latest pack version of the Forensic Tool with adaptive prerequisite".

True to its technological roots, the new self-managing industry, press and suppliers have come up with several different names for autonomic computing and related technologies. This is likely to cause confusion to management and end users alike. This strengthens the argument for immediate standardization, before the industry becomes mature. Figure 7 highlights a few examples of companies and their standards for autonomic computing.

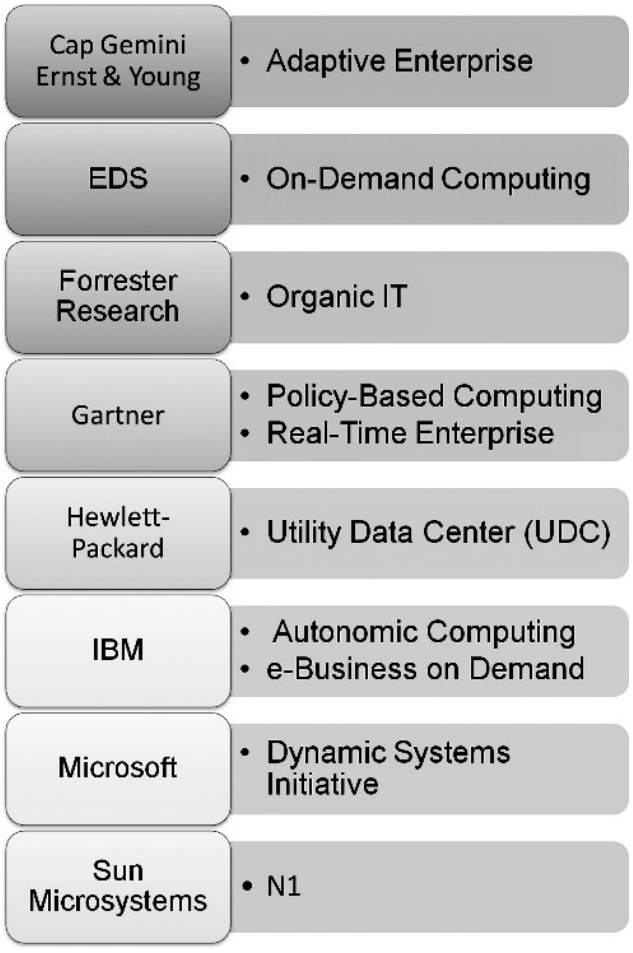


Fig. 7. Some companies and standards for autonomic computing

3. Conclusions

The main objective of this work was use of autonomic mechanisms to support computer crime investigation management in police forces and forensic organizations. We did a roadmap conceptualization of this new knowledge area achieving the first steps towards the autonomic technology in usual forensic application domain.

This paper also proposes an ontology-based knowledge representation (Autonomic Forensic Ontology–AFO) to solve the problem of knowledge representation in forensic computing. The AFO presented here aims to support the formal modeling of knowledge for the forensic environment and is mainly concerned with the study and understanding of this environment. Also, we implemented ontologies in some case studies for computer crime investigations using Protégé language.

In addition, we have proposed a novel autonomic management architecture applied to forensic domain, called AFoMA. It introduces different techniques to manage and integrate heterogeneous and distributed computing resources using five layers of control. The AFoMA architecture brought a set of new contributions:

1. The use of AFO ontologies that augment the knowledge contained in the model to better define semantics and meaning of facts, rules, norms, and contracts.
2. The use of an inference machine that can make deductions not explicit in knowledge repositories. This machine is capable of developing to the point where it can treat more elaborated elements as rules and norms that formed the knowledge and the learning
3. The use of self-management elements: self-configuration, self-optimization, self-healing, and self-protection.
4. The use of an extensible knowledge and learning manager that is independent of application.

5. The use of multi-agent systems to facility code mobility, helpful, rationality, adaptability, and collaboration.

Future work will concentrate on development of a complete realization of the architecture, which will implement each layer of our architecture incorporating autonomic capabilities.

References

[1] P. Horn, *Autonomic Computing: IBM's Perspective on the State of Information Technology*, 2001. http://www-03.ibm.com/autonomic/library.html. Last access in 7th June 2008.

[2] IBM, *Practical Autonomic Computing: Roadmap to Self Managing Technology*, 2006. http://www-03.ibm.com/autonomic/library.html. Last access in 7th June 2008.

[3] Intel, *Standards for Autonomic Computing*, Intel Technology Journal, Volume 10, Issue 04, 2006. http://developer.intel.com/technology/itj/index.htm, Last access in 7th June 2008.

[4] Motorola, *Autonomic Architecture to Support Next-Generation Services*, 2008. http://techpubs.motorola.com, Last access in 19th June 2008.

[5] M. Parashar, and S. Hariri, *Autonomic computing: concepts, infrastructure, and applications*, CRC Press, 2007.

[6] G. G. Matthews, *Neurobiology: Molecules, Cells, and Systems*, Blackwell Pub, 2nd edition, 2001.

[7] G. A. L. Campos, A. L. B. P. Barros, J. T. Souza, and J. Celestino Jr. *A Model for Designing Autonomic Components Guided by Condition-Action Policies*, 2nd IEEE Workshop on Autonomic Communications and Network Management – ACNM, Salvador: Brazil, 2008.

[8] Motorola, *Philosophy and Methodology for Knowledge Discovery in Autonomic Computing Systems*, 2008. http://techpubs.motorola.com, Last access in 19th June 2008.

[9] M. Esseghir, S. G. Doudane, K. Haddadou, *First Steps Towards an Autonomic Management System*, in Proceedings of the 11th IEEE/IFIP Network Operations and Management Symposium – NOMS, 2008.

[10] J. O. Kephart, and D. M. Chess, *The Vision of Autonomic Computing*, IEEE Computer, January, 2003, pp.41–50.

[11] IBM, *An Architectural Blueprint for Autonomic Computing*, 2006. http://www-03.ibm.com/autonomic/library.html. Last access in 7th June 2008.

[12] T. Gruber, *A Translation Approach to Portable Ontology Specifications*, Knowledge Acquisition, Vol. 5, 1993, pp.199-220.

[13] J. H. M. Nogueira, *Ontology for Complex Mission Scenarios in Forensic Computing*, In proceedings of the 2nd International Conference of Forensic Computer Science, Guarujá: Brazil, 2007

[14] M. J. Almeida, F. Freitas, R. R. Azevedo, and Dias G, *An Ontology for Information Security Management in Autonomic Computing Environments*, in Proceedings of the 2nd Latin American Autonomic Computing Symposium - LAACS, Petrópolis: Brazil, 2007.

[15] E. Lehtihet, J. Strassner, N. Agoulmine, and M. O. Foghlú, *Ontology-Based Knowledge Representation for Self-Governing Systems*, Lecture Notes in Computer Science, Vol 4269, Springer-Verlag Berlin Heidelberg, 2006, pp.74–85.

[16] Chimaera, *Chimaera Ontology*, 2007. www.ksl.stanford.edu/software/chimaera. Last access in 23th June 2008.

[17] Ontolingua, *Ontolingua System Reference Manual*, 2007. http://www-kslsvc.stanford.edu:5915/doc/frame-editor/index.html. Last access in 23th June 2008.

[18] Protégé. *The Protege Project*, 2007. http://protege.stanford.edu. Last access in 23th June 2008.

[19] Protégé. *Using Protégé-2000 to Edit RDF*. Technical Report, Knowledge Modelling Group, Stanford University, 2006. http://www.smi.Stanford.edu/projects/protege/protegerdf/protege-rdf.html. Last access in 23th June 2008.

[20] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2nd edition, 2002.

[21] S. Casare, *Uma Ontologia Funcional de Reputação para Agentes* (in Portuguese language) PhD Thesis, USP University, São Paulo: Brazil, 2005.

[22] J. Ferber, O. Gutknecht, and M. Fabien, *From Agents to Organizations: an Organizational View of Multi-Agent Systems*, in P. Giorgini, J. P. Muller, J. Odell editors, 4th International Workshop on Agent Oriented Engineering. LNCS 2935, Berlin: Springer-Verlag, 2003.

[23] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman, *Creating an Ambient Intelligence Environment Using Embedded Agents*, IEEE Intell Systs, 2004.

[24] J. H. M. Nogueira, *Using Agents to Support Mission-Critical Scenarios*, Seminar, Department of Computing Science, University of Aberdeen, 2007. www.csd.abdn.ac.uk/research/seminars/seminar.php?id=196. Last access in 20th June 2008.

[25] J. H. M. Nogueira, *Mobile Intelligent Agents to Fight Cyber Intrusions*, International Journal of Forensic Computer Science, Brasília: Brazil, 2006.

[26] O. Boissier. *Modeles et architectures d'agents* (in French language), Principes et architecture des systèmes multi-agents, chapter 2, J. P. Briot, and Y. Demaezau, Ed. Paris: Hermes, 2002.

[27] W. Vasconcelos, M. McCallum, T. Norman, *Modelling Organisational Change using Agents*, Technical Report AUCS/TR0605, Department of Computing Science, University of Aberdeen, 2006.

[28] J. Strassner and D. Raymer, *Implementing Next Generation Services Using Policy Based Management and Autonomic Computing Principles*, in Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium – NOMS, 2006.

[29] Motorola, *Autonomic Systems and Network - Theory and Practice*, 2006. http://techpubs.motorola.com. Last access in 19th June 2008.

[30] Motorola, *Knowledge Management Issues for Autonomic Systems*, 2008. http://techpubs.motorola.com. Last access in 19th June 2008.

[31] W. Vasconcelos, *Norm Verification and Analysis of Electronic Institutions*, vol. 3476, LNAI. Berlin: Springer-Verlag, 2004.

[32] A. García-Camino, J. A. Rodríguez-Aguilar, C. Sierra, and W. Vasconcelos, *A Distributed Architecture for Norm-Aware Agent Societies*, vol. 3904 of LNAI, Berlin: Springer-Verlag, 2005.

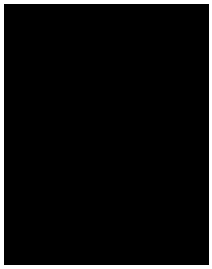
[33] J. Vázquez-Salceda, H. AldeWereld, and F. Dignum, *Implementing Norms in Multi-agent Systems*,vol 3187, LNAI, Berlin: Springer-Verlag, 2004.

[34] Intel, *Towards an Autonomic Framework: Self-Configuring Network Services and Developing Autonomic Applications*, Intel Technology Journal, Volume 08, Issue 04, 2004. http://developer.intel.com/ technology/itj/index.htm. Last access in 20th June 2008.

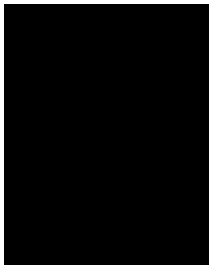
[35] J. Strassner, *Autonomic Networking - Theory and Practice*, in Proceedings of the 11th IEEE/IFIP Network Operations and Management Symposium – NOMS, 2008.

[36] is missing;

[37] ic computing



José Helano Matos Nogueira
Technical-Scientific Sector, Brazilian Federal Police, Fortaleza – Brazil, helano@apcf.org.br



Joaquim Celestino Júnior
LARCES/UECE, State University of Ceará, Fortaleza – Brazil, celestino@larces.uece.br

IJoFCS (2009) 1, 43-49

e-forensic PRESS

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

Computer Assisted Systems For Forensic Toxicology

William Alan Alexy

Cuyahoga County Coroner Systems Analyst

Abstract - We report the development of a computer hardware and software that addresses the special needs of the forensic toxicology laboratory for real-time, data gathering, analysis, and retrieval. In addition to accessioning, work-list preparation, and result reporting, we implemented automatic test ordering based on patient/decedent case characteristics in the accessioning stage to provide reliable and uniform analyte profiles for case solutions. The system also provides extensive real-time event journaling in order to satisfy strict chain of custody requirements consistent with the College of American Pathologists Accreditation, Substance Abuse and Mental Health Services Administration (SAMHSA) certification and the American Society of Crime Lab Directors (ASCLD).

The relationships among analyte concentrations in various specimens (e.g. blood, urine, gastric,) have been incorporated into the software as functions of time before and after death before the next step in the case life-cycle. These recommended tests are continually reviewed to educate and verify lab procedure. The system has saved many hours of error-prone manual work. It has streamlined data collection and made a broad spectrum of expertise available to the laboratory at all times. These features have decreased error rates, increased productivity and enhanced the forensic skills of the laboratory.

KEY\WORDS - toxicology, information systems, computers, laboratories, accessioning, toxicology laboratories, expert systems, epidemiological.

1. Introduction

This communication describes the design and implementation of a highly successful information gathering retrieval and analysis system used in the toxicology laboratory of the Office of the Cuyahoga County Coroner (OCCC) from 2003 to 2008. For many years our laboratory was concerned about potential deficiencies or omissions with respect to chain of custody recording, erratic test ordering, lost or delayed data, transcription errors, and all of this with an ever increasing workload. (The implementation of systems to address other information management problems in the areas of primary drug standard inventory, general supply inventory, purchasing and receiving and equipment maintenance will be addressed in subsequent communications.)

The OCCC serves the greater Cleveland area. This jurisdiction encompasses a population of 1.5 to 2.0 million people whose average mortality rate is 1% (15,000 to 20,000 deaths a year). Approximately 20% of these deaths become OCCC cases (3000 to 4000 cases a year). The OCCC also offers a referral service to coroners from surrounding counties (40 to 60 cases per year), and serves local police jurisdictions by testing individuals suspected to be driving under the influence (DUI/DUID) (up to 400 per year). We accept biological samples from other forensic agencies and area hospitals for comprehensive drug screening in alleged poisonings and therapeutic drug monitoring (TDM) applications adding another 300 cases per year. Since 2000, the toxicology laboratory of OCCC has processed over 35,000 tests per year using TOXLAB02. Due to this heavy workload we found it necessary to design a more efficient system. We are presenting TOXLAB02 here for others that may be interested in adopting it.

2. General Design Considerations

We designed a system that could be maintained by the budget constraint of a single technical staff member, with minimally a bachelor's degree in computer science. We chose Microsoft's operating systems platform because it could be regularly upgraded and is compatible with the systems of major toxicology laboratory instrument manufacturers. For instance, Linux an open-source operating system was not used because of security concerns. Our desire was to design a system on a platform with proven reliability and functionality.

3. General Goals and System Specifications

The basic system requirements were defined and alternatives for meeting the goals were studied. A primary concern of the design was the incorporation of over 60 fundamental and commonly required tests such as GC/MS, GC's, HPLC, Chem. 7, Immunoassay and ELSIA (enzyme-linked immunosorbent assay). To our knowledge, no software with these capabilities was commercially available. Six general goals and several system-specific requirements were developed:

1) A centrally managed, interactive, real-time data sharing environment that permits intra- and inter-departmental access to common data. This would reduce duplication of efforts in transcription and excessive paper usage.

2) A system that maximizes the efficiency and time management of professional and technical staffs to (e.g. minimizes the technical staff's clerical duties).

3) A system that catalogues all events that transpire while a specimen is in our custody. This is a special need essential to the forensic science laboratory.

4) A system that improves the tracking of in-process specimens to minimize the time required to determine status and to increase specimen throughput.

5) A system that permits the construction of real-time interfaces for data acquisition from laboratory instruments and personal computers. We also considered the allocation of jobs to persons and machines to maximize the strengths of each. The current lab structure employs networkable GC/MS, GC's, HPLC, Chem 7, Immunoassay and ELISA instruments. Networkable computer workstations are also available to all staff members for case work.

6) A system that meets the laboratory procedure standards of the American Society of Crime Laboratory Directors (ASCLD), with initial capital costs for hardware, software and initial development that do not exceed $400,000.

Specifications

The toxicology laboratory occupies 482 m^2 (4736 ft^2) of a single floor. It is staffed by two Ph.D.s, seven forensic science professionals, a

secretary, and a laboratory aide. The laboratory uses an integrated analytical approach for the investigation of alleged illegal activity. To organize testing the laboratory has been divided into thirteen groups, with the ability to be expanded into twenty. Testing employs the appropriate standards and control proficiencies for GC/MS, GC's, HPLC, Chem 7, Immunoassay and ELISA screening. We are presently using 14 of these instruments and 14 workstations.

4. System Selection

Criteria, requirements, design and limitations

We sought to design an application--specific programming module set that encompassed the current job areas of the laboratory: accessioning (the acquisition of sample material), chemist (the first level of drug analysis), supervisor (reviews chemist results and assigns new or re-testing), manager (reviews supervisory results, monitors work load and recommends further testing along with reporting functions), invoicing (to outside agencies) and medications (case drug inventory).

The system software was developed according to the following requirements: (1) software that allowed maintenance by outside consultants; (2) gradual development of software so that the operating environment could be tailored to the needs of the coroner's office; and (3) compatibility with all hardware platforms (e.g., Java, VM Ware or X-86). Programs were developed and tested on a project-by-project basis to allow ample time for refinement of the applications. Stringent requirements for the system software primarily included maturity of a stable platform.

Two additional requirements were the ability to run commonly used commercial spreadsheet, word-processing, and statistics software and production of data compatible with the systems of outside agencies. The system was also required to allow integration of all major hardware elements into a commercial secure local area network.

Three fundamental approaches to meeting the system specifications were considered during the design process. The first was a single personal computer (PC) the second was a network of PCs and the third was a centralized server computer.

The single PC option was not ruled out because its most common operating system (Windows XP) supported up to ten concurrent users, it had adequate backup capability and a low cost. Previous generations of the PC could not claim the versatility of scalability. PCs have substantial quality, durability, and expandability and provide adequate mass storage backup with limited expense. The PC software market is also rapidly catching up with previous generations of software development for main-frames. In fact, the efficient design of migration paths from older hardware and software to new versions has become a concern of PC developers. High-quality PC networks have matured in concert with reliable networks and have been used with mini-computers for many years.[6] For these reasons, single PCs and PC networks became the core background for the development of our systems.

In designing the development path, we first developed a single PC-based system, then a version of the same system with multi-user capability. Second, we developed a server with a smart-client-based system. Next, we implemented a highbred virtual server, a smart-client system for a larger decentralized laboratory. We realized that the second generation had been designed by few and the third generation has not been built by anyone. The first generation was named 'Toxlab02', the second is named 'Pathways' and is in beta testing at the OCCC. The third generation is in the development stages at the OCCC.

The goals set above were most economically achieved by a centralized mid-range PC with network expansion potential. A PC such as the Dell Precision 360 has high-speed processors and a wide range of peripheral hardware support, including multi-Terabyte disk arrays, DVD backup and full and mature network connectivity. These systems are well supported by mature multi-

user, multi-tasking operating systems, database packages, and network products.

Microsoft Access was chosen for data management because it parallels XP platform development. In addition, Access has the ability to migrate to Microsoft server software using the file server component.[1]

To satisfy the need for special logic experts and automated accessioning experts, high-level language software was integrated into the system. The accessioning expert creates a list of tests to be performed on a case based on a combination of three factors (type of samples, type of case and manner of death). The automated function tracks development of testing ordered by these three factors, and lets the accessioned know which testing is currently being prescribed by the lab manager. It serves to eliminate errors due to forgetfulness based on combinations of sample types, case types or manner of death parameters. This requirement could only be fulfilled through a mature, well-supported high-level language program such as some components of the Microsoft Visual Studio Suite.

The lab needed a combination of products that objectively exceeded our expectations, that implementation then could meet our subjective requirements. This was manifested in a method that avoided any tendency to 'profile' a case. Profiling presupposes the nature of the development path of artificial intelligence (AI) experts. An AI expert allows a system to evaluate current work performance, and make suggestions to the chemist based on these results. In addition, an AI expert will "learn" and thus improve functionality over time. The system shell used for AI was word process able in order to increase efficiency. The 'learning from self' became impracticable for half of the project as the complexity of the experts became specifically known.[2]

The appropriate operating system software had to be multi-user and multi-tasking. The mature real-time Microsoft operating system XP (Experimental) was chosen. This system could support the concurrent use of 10 terminals and 10 printers for data entry and lookup functions. In addition, it is accompanied by a rich set of simple instructions and has a user-friendly reputation. Log-in is controlled by passwords that can be changed by the user or the system manager, and can be set to expire. In addition, all application programs access a common self-developed security system that prevents usage by unauthorized persons and limits the use of the programs to specific users.

With this configuration, the manager is able to produce professional-quality final documents. Transmission of the documents is not permitted in order to maintain the lab's level of security and control of its products.

5. Start-Up: Operational Experiences

Hardware installation took about two days. Installation of cable required 200+ man-hours. The operating system was installed by the hardware vendor. Initial installation of the software was accomplished by an in-house network specialist. Thereafter, software loading was automated and any user could update software from the server at any time.

Users had requested the ability to automatically print cases from within application programs. While automation was not possible, a user may issue a "click print" command to any system printer, including networked printers in the local area network. Many programs allow users to select a specific printer to use for their output. Batch processing of case load and results is easily accomplished with this system. The user also receives automated chain of custody in a case folder and in printed form for use in the lab. Batch work and chain of custody forms are held in a library of Microsoft Excel forms. These forms can be modified, added or deleted to meet the needs of new instruments and changes in the laboratory environment. All forms are saved as part of chain of custody for each technician.

We chose to develop the first generation laboratory applications in Visual Basic 6.0, which

allows Microsoft Access databases to be read or written with batch programs. Total development time for the specifications for all the applications was approximately 6000 man-hours. Each application was programmed, tested, modified, and retested until it achieved production quality. At that time, parallel trials were performed against the older system. When these trials were completed, the programs were not permitted to operate in the production environment until a change mechanism had been implemented to migrate from the previous generation of software.

All applications developed for the toxicology laboratory used two strategies to minimize development effort and time, and maximize data availability. The first was the use of subroutines further amplified by the use of Graphical User Interface (GUI), and standardized program logic Gang of Four (GoF) design patterns that could be easily incorporated into new applications. The second strategy was the use of the SQL relational database, which permits the development of applications or segments of applications, somewhat independently, while guaranteeing that any application can access and display data from any other application. The data are shared by referencing a common key such as case number in the data dictionary of each database. Selected information requests from many independent sources can thereby be collected and used together.[53]

The TOXLAB02 applications were completed in a period of fourteen months. During this time the laboratory staff was trained to use the system. The staff also contributed to the specifications and design of the software. With the completion of each new module the programmer prepared technical and user documentation, in cooperation with the director. The documentation process required 1000 man-hours and further improved the efficiency and functionality of the programs. When the effort to establish this computer system began, the OCCC toxicology laboratory was a member of the American Society of Crime Lab Directors (ASCLD); the documentation, chain of custody, validation processes and reports were designed to meet this organization's requirements for certification.

Programs may be run on any PC in the laboratory by clicking on the desired module or through a GUI multi-set interface. Several persons may simultaneously use the same program. Data locking features use the pessimistic model, which allows a toxicologist to change information on a case presently viewed by another toxicologist. In the event of an update by the first viewer, the second viewer is advised that information has changed and given the option to update the information by reloading the screen.

The TOXLAB02 program permits new cases to be entered with demographics, case detail and a specimen list. The program automatically generates test orders via (AUTO-ORDER-EXPERT) based on several variables. Each entry is placed in the chain of custody record, in real time, to provide an audit trail of activity.

Chemist/supervisor/manager modules permit scientists to alter any data item in a database master record and to change automated orders, depending on their security. All changes, deletions and additions to these programs are journaled as part of the chain of custody for each case, to provide an audit trail of activity.

The work-list module assembles a bench-by-bench work-list of all cases for which work is pending. Printouts use bold-face fonts to alert the toxicologist to cases that are urgent and/or infectious in nature. All cases are printed in reverse chronological order, so that older cases appear at the top of each individual bench-group worksheet. A work-list may be run on demand by the user and may also be automatically scheduled.

The results module is the main result entry and modification program. It contains a large expert system. Toxicologists may select individual cases through batch mode, declare results, and receive notification of any tests that need to be scheduled based upon these results. This module is dependent on the work-list module. The work-list module organizes samples by type and test;

this facilitates entry of results (i.e., positive, negative, quantity not sufficient and unsuitable findings) with a single key stroke. Qualitative and quantitative results may be changed or deleted before the supervisor's review and approval of the case. The appearance of certain sample-specific results will cause the program's expert to automatically create new test orders for the same or other samples. When all non-negative results have been entered, the toxicologist declares the case to be "ready" for the supervisor's review, in the computer. Every result is journaled in real time to provide an audit trail of all activity.

The manager module permits the lab manager to find and review all cases for which all ordered results have been completed. If the lab manager approves the case, it is internally identified as "approved" and ready for printing. This event is journaled in real time and provides the desired audit trail to verify results for toxicological consistency. This module also prints final reports of cases previously approved by the supervisor. The reports cannot be changed in any way, except by a manager using the manager program. All final reports generated in the laboratory are printed and journaled.

Epidemiological research is the net result of creating a data structure of this type. These programming tools are certain to create a bright future in Toxicology with the use of computers.

6. End of Life Cycle

The life cycle policy is designed to specifically determine the length of time for product management and to schedule development of an updated product. In general, a minimum of 10 years of support is accepted for computer software. The system discussed in this article has been operational for seven years. The life cycle combines two time periods: five years of main vendor support or two years after the successor product (N+1) is released, whichever is longer; and five years of extended support or two years after the second successor product (N+2) is released, whichever is longer. We concur with these standards and promote the knowledge gained from previous generations of software development.[4]

7. Conclusions

The implementation of computerization in the forensic laboratory is a foreseeable extension of the field of Toxicology with the advent of inexpensive computer systems. As we know the software has been developed for a forensic laboratory which does extensive testing – far beyond the pale of the standard "alcohols" type lab. The accuracy of the information, the speed in which it is available and the security involved, became challenging road marks to meet and exceed.

The computer has become a valuable tool that when appropriately applied can and has produced a cost-effective efficient and accurate aid to the forensic toxicologist. The system we have had in operation for the last 7 years has provided us with an operational tool and a source for looking back at the foot prints in the sand to see what we have done.

8. Acknowledgements

We would like to thank all of those who helped make the first version of the Toxicology software a working success:

Dr. Frank P. Miller, MD Coroner, Cuyahoga County, Ohio,
Dr. Elizabeth K. Balraj, MD, Cuyahoga County, Ohio,
Eric Lavins, B.S., Cuyahoga County Coroner,
Troy Merrick, M.S.F.S., Cuyahoga County Coroner,
Szabolcs Sofalvi, M.S.Ch.E., Cuyahoga County Coroner.

References

[1] Access 2003, Allison Balter, Sam's Publishing, Indianapolis, 2004.

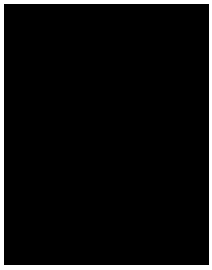
[2] Evaluation of Isopropanol Concentrations in the Presence of Acetone in Postmortem Biological Fluids, Amanda Jenkins, Journal of Analytical Toxicology Niles, IL. 2008.

[3] SQL Server 2005, Dejan Sunderic, MCDBA, McGraw-Hill, 2006.

[4] Upgrading to .NET, Ed Robinson, Michael Bond, Robert Oliver, Microsoft Press, Redmond, WA, 2002.

[5] Design Patterns in C#, Stephen John Metsker, Addison-Wesley, Boston, 2004.

[6] Internet information Services (IIS) 6.0, Microsoft Team, Microsoft Press, Redmond, WA, 2004.



William Alan Alexy
Cuyahoga County Coroner Systems Analyst

IJoFCS (2009) 1, 50-57

e-forensic PRESS

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

A New Digital Evidence Retrieval Model for Gambling Machine Forensic Investigation

Pritheega Magalingam(1), Azizah Abdul Manaf(2), Rabiah Ahmad(3), and Zuraimi Yahya(4)

(1) Centre For Advanced Software Engineering, mprithee@gmail.com
(2) College Science and Technology, azizah07@citycampus.utm.my
(3) Centre For Advanced Software Engineering, rabiah@citycampus.utm.my
(4) Faculty of Electrical Engineering, Universiti Teknologi Malaysia, International Campus, Jalan Semarak, 54100 Kuala Lumpur, zuraimibinyahya@yahoo.com

Abstract - Gambling machines serve as the principal means by which illegal games are conducted. This paper presents a method for retrieving information from seized gaming machines along with an analysis of the interpreted information to prove that the gaming machine was used illegally. The process is illustrated using a machine seized from a suspected illegal gambling operation. A detailed gambling machine forensic procedure provides important assistance to forensic investigators (e.g., police or private investigators) in gathering evidence relevant to illegal gambling.

Keywords - digital forensic, forensic analysis, gambling machine, information retrieval, digital evidence, interpretation, string search

1 Introduction

Any device used for calculation, computation, or information storage may be used for criminal activity, by serving as a convenient storage mechanism for evidence or in some cases as a target of attacks threatening the confidentiality, integrity, or availability of information and services. Computer forensic analysis focuses on the extraction, processing, and interpretation of digital evidence.

A major challenge for police forces is determining whether gaming machines in cyber cafes are being operated illegally. Modern gaming machines contain sophisticated computer software and hardware, and locating relevant digital evidence becomes a difficult task requiring the assistance of forensics experts. Presenting this evidence in court requires a detailed analysis of the gaming machine hardware used to store data and programs, a method of extracting data from non-volatile memory, and an examination of the data to obtain reliable evidence.

2 Background Problem

Technological evolution has enabled computers to serve as gambling machines in which all functions are electronically controlled. Some

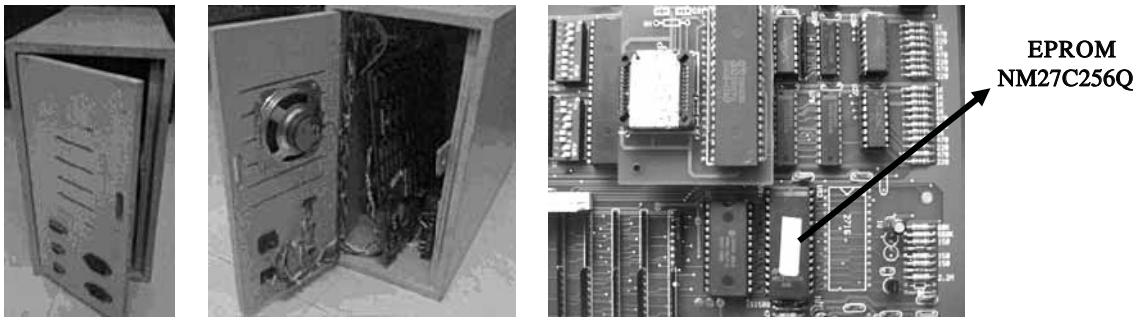


Fig. 1. Gaming CPU and Z80 Microcontroller

gaming machines are constructed with a motherboard programmed to provide a dual function, allowing players to use the machine for amusement or for gambling. Switching the operating mode is a common method of preventing the police from discovering illegal gambling [1].

The major functions of the machine are controlled via software encoded on a non-volatile EPROM (Erasable Programmable Read Only Memory) chip [1]. Older machines may be converted into amusement machines by inserting new EPROM chips. However, if the EPROM is programmed for gambling functions, the device may be operated illegally. A gambling machine serving as physical evidence in a court case represents a significant challenge in a computer crime investigation. Unfortunately, the current procedures for identifying CPUs containing games related to gambling are inadequate because it is difficult to visually differentiate a normal operating system from a system running a gaming event.

This paper provides guidelines for determining whether confiscated computer devices contain useful evidence and proposes evidence acquisition and examination procedures. The procedures were employed in a hardware forensic analysis conducted on a gaming machine manually assembled by the owner and seized by the Royal Malaysian Police Force. Figure 1 contains photographs of the seized gaming box and the EPROM installed in the printed circuit board.

3 Computer Forensic Research

This paper focuses on computer misuse and methods of acquiring digital evidence from electronic machines. Unlicensed gaming devices fall into the category of computer misuse when they are used to conduct illegal gambling [2]. The components of an electronic device should be traced in order to obtain investigative information[3], and the memory should be analyzed. Several previously published memory acquisition procedures for microprocessor-based devices are described below.

3.1 Forensic Data Recovery from Flash Memory

Marcel Breeuwsma *et al*. (2007) claimed that most forensic tools currently on the market perform logical data extraction and are not capable of retrieving all possible information from the storage medium. Three methods for low-level information acquisition from flash memories were introduced, including flasher tools, the use of an access port for testing and debugging, and a semi-invasive method in which flash memory chips were physically removed from the printed circuit board [4]. The paper also described the steps necessary to translate the extracted data to the file system level. Our exhibit falls into the third category, since the seized wooden gaming box contained a printed circuit board with no means for external connections.

3.2 Memory Acquisition Procedure for Digital Investigation

A hardware-based procedure for information retrieval from volatile memory was described by Brian D. Carrier and Joe Grand (2004), who also claimed that existing data acquisition methods are unreliable because they write back to the memory and use only certain tools to obtain obvious data

(leaving the rest of the memory unanalyzed). Their solution was to install a Peripheral Component Interconnect (PCI) expansion card before the crime occurs. The back of the card is equipped with a switch to activate the PCI controller on the card. Once activated, the card takes control of the PCI bus and is able to access memory without relying on the operating system or system memory for storage. It will copy the exact contents of the volatile memory to an external non-volatile storage medium [5].

3.3 Xbox Forensics

Burke and Craiger [20] reported an easy and non-intrusive method of data extraction to identify whether hackers have compromised an Xbox by installing non-approved software to run an operating system other than the one originally installed. The author used Linux to conduct the Xbox analysis, and the output was examined line by line. The use of the string utilities and hex viewer in Linux provided a good starting point to determine if evidence existed on the partition in ASCII form and helped to describe the binary data in the retrieved evidence.

3.4 Forensic Investigation of Nintendo Wii

The Nintendo Wii is a gaming console offering 256MB of flash-based memory that can be wirelessly connected to the internet. Dr. Benjamin Turnbull [7] aim of this investigation was to record all activity to ensure the system was unaltered. This gaming console features automated logging, which records information including the game being played and the duration of play. The investigation method involved activation of an external logging mechanism or recording device, determination of the current unit time in the system settings, and identification of the messaging system and the extent of system use, i.e. notes sent between individuals on a particular date [7].

3.5 A Methodology for Forensics Analysis of Embedded Systems

Kyung-Soo Lim and Sangjin Lee [21] introduced a two-phase analysis method for embedded systems such as Microsoft Xbox, Sony Playstation 3, Nintedo Wii, and GPS navigation units. In both phases, the authors compare the target system information with information provided by the manufacturer to identify illegal activities. In our case, the seized gaming machine was not built by a specific manufacturer but by the owner of a cyber café, and specific examination of the chips connected to the existing microcontroller was essential.

4 Forensic Analysis Design

In order for a gaming machine to be classified as an illegal gambling machine, the evidence must support certain facts, and the following three relevant pieces of information must be present [10]:

(a) A betting mechanism that allows the raising of various sums of money depending on the outcome of the game,

(b) A random number generation process to establish the game results, and

(c) A payout value displayed to winning players.

The information may be extracted from the EPROM program memory embedded in the gaming machine microcontroller [11]. Relevant information concerning this process was gathered from optimal practices as well as standard operating procedures. A proposed evidence retrieval method is diagrammed in Figure 2.

5 Implementation and Results

5.1 Evidence Acquisition

Fortunately, in our case the EPROM was inserted into a chip socket and could be gently removed from the microcontroller board using forceps. This is a better method than de-soldering, since the heat required for de-soldering may damage the memory chip. The test may only be performed on non-encrypted EPROMs. The type of EPROM being examined was the NM27C256Q, and the ChipMax reader was selected for program extraction [13].

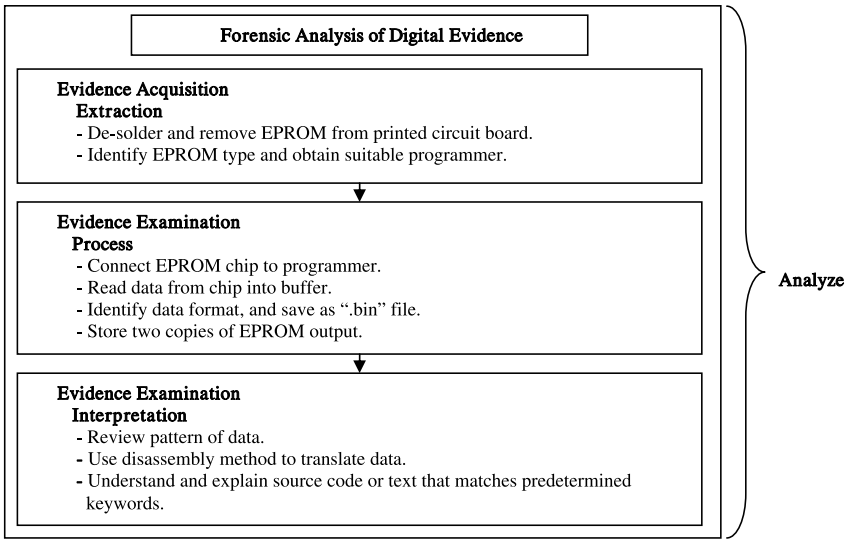


Fig. 2. Gambling Machine Forensic Analysis Guidelines

Figure 2 lists the steps involved in digital forensic investigation, emphasizing the evidence analysis phase. This phase is divided into evidence acquisition and examination activities, and appropriate guidelines are mapped onto each main step.

5.2 Evidence Examination Procedure

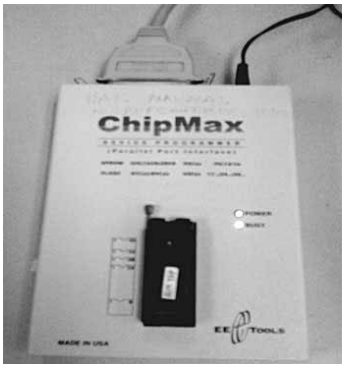


Fig. 3. EPROM in ChipMax Reader Socket [13]

5.2.1 Process

The ChipMax reader depicted in Figure 3 was used to read the EPROM memory. The EPROM was placed in the reader socket and two copies of the EPROM contents were saved in binary (".bin") form. One copy was kept as original evidence and the other was used during the forensic examination. A hash value generated from both copies was used to demonstrate that the evidence had not been modified.

5.2.2 Interpretation

Reverse engineering is the process of translating the object code into understandable source code [12]. Several software tools were identified and tested for use in disassembly and conversion to source code, including the Barleywood Z80 Simulator, Z80 Simulator IDE 8080, and Z80 Assembler Disassembler Suite. The most suitable tool in this case was the Z80 Simulator IDE. Representative output of the disassembly process appears in Figure 4.

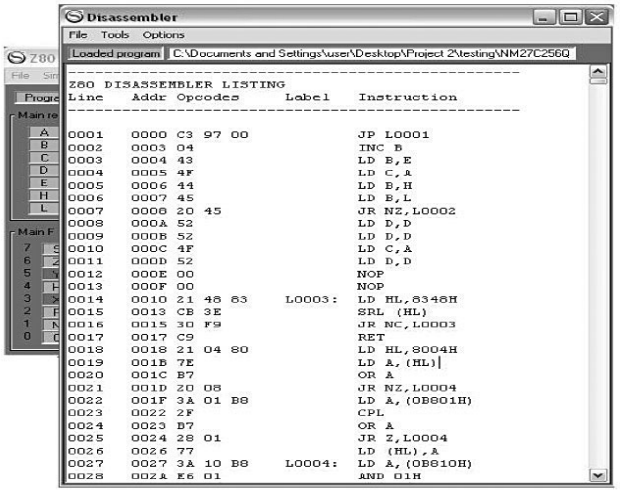


Fig. 4. Output of disassembly process

6 Output Analysis

The output following the disassembly process is an assembly language program, which in our

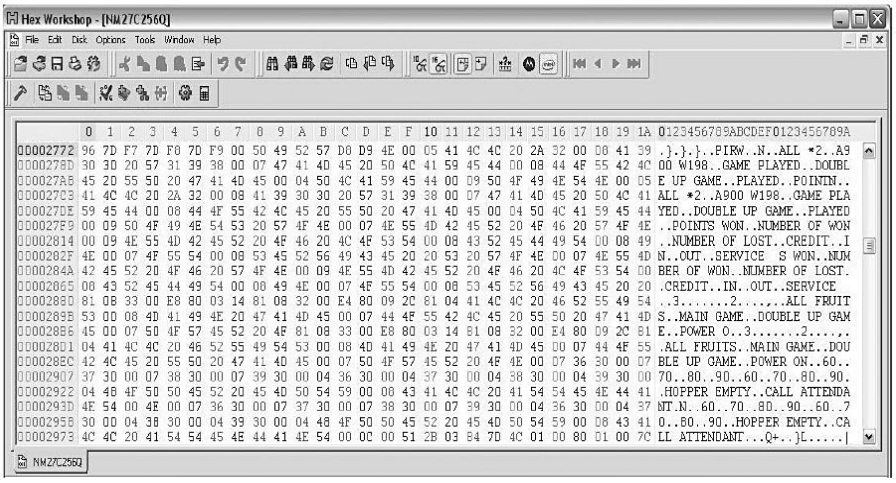


Fig. 5. Gambling terms embedded in machine memory

case study was written using the Z80 instruction set. Each command in the program was analyzed to facilitate understanding of the program function [8]. In particular, the program analysis sought to identify common gambling functions such as a random number generator, betting function, and payout mode. Several factors hampered the identification of subroutines involved in the gaming operation, including frequent repetition of the Z80 instruction pattern and an inability to determine the starting instruction point in the absence of inputs from the actual gaming hardware. For this reason the assembly program was trapped in a loop for almost 12 hours.

In order to circumvent these problems, we developed an alternative analysis method. Data related to the current game or to the last game played could provide the game sequence and output [14]. The machine code retrieved from the EPROM was therefore converted into readable plain text to enable string searching using a hex editor.

During the machine code analysis, a group of symbols, numbers, and letters related to gambling operation were identified. These lines were collected to determine their actual meaning, leading to the discovery of additional gambling terms:

*(a) ALL *2..A900 W198..GAME PLAYED*

"ALL" in the statement shows that the player chose to play all games. This means that the machine will rapidly display all of the games available based on the player's purchased ticket [15].

(b) DOUBLE UP GAME

The term "double-up game" indicates that the player chose to play [16] a second time using a wager amount equal to the value played in the first round. "Wager" indicates the amount the player paid to play the game. Wagers typically take the form of a token, coin, or currency note.

(c) POINTS WON..NUMBER OF WON.. NUMBER OF LOST..CREDIT..IN..OUT.. SERVICE S WON..NUMBER OF WON.. NUMBER OF LOST..CREDIT..IN..OUT.. SERVICE

This information pattern stored in the EPROM indicates that the player activated services within the machine to view the number of points won, number of points lost, value of money credited, and credits won.

(d) ALL FRUITS

This is the combination of composite symbols on the gaming machine monitor that represents a winning combination [15].

(e) MAIN GAME..DOUBLE UP GAME..POWER ON..60..70..80..90..60..70..80..90

This indicates that the player returned to the main game and doubled up the betting amount

(i.e., he increases the money to play another set of games). The numbers "60..70..80..90.." could represent the winnings during play.

(f) HOPPER EMPTY..CALL ATTENDANT

This string is used to report a fault condition in the coin output (hopper) system when the player attempted to redeem the money won through the game. The message is displayed if the payout coins did not pass a hopper output sensor within a specified time [14], and instructs the player to receive payment from the attendant.

(g) SPECIAL ODDS FOR TOTAL BET

According to Casino Gambling Terms and Definitions, the "odds" describe the ratio of probabilities or the amount a bet pays [17]. The pay-out table holds the combinations of game elements that will appear in the video cells and the pay value is associated with a winning combination of game elements [18]. The probability table or pay-out table is stored in the EPROM and is accessed by the odds routines to calculate the points won by the player [9].

8 Contribution

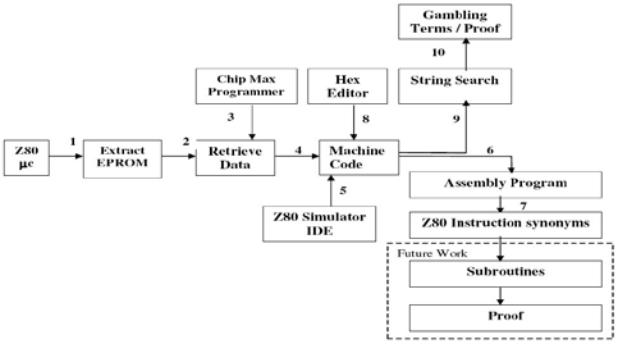


Fig. 6. Information Retrieval and Evidence Analysis Model

This study has contributed to the development of a gambling machine forensic analysis model. Figure 6 diagrams the information retrieval and evidence analysis process. Each arrow in the diagram is numbered and represents a certain function involved in the forensic analysis process.

{1}: The EPROM chip is removed from the Z80 microcontroller.
{2},{3}: Information stored in the EPROM chip is retrieved using Chip Max programmer.
{4}: The output from the chip reading process is identified as machine code.
{5}: The Z80 Simulator IDE is used to disassemble the machine code.
{6}: The output from the disassembly process is examined.
{7}: The assembly program is translated manually into Z80 instruction synonyms.
{8}: The machine code is read using Hex Editor tool.
{9}: The output from step {8} is a group of symbols, numbers, text and letters.
{10}: A string search process is conducted and gambling terms found.

9 Conclusion

A gaming machine is considered a gambling machine when it involves an actual monetary transaction and a bet to win the game. In order to choose the winning combination, a random number generating process is called and the payout value is selected from the payout table stored in memory. The following routines are commonly identified in gambling programs, and were present in our case study:

(a) A betting mechanism

The "double up game" string identifies a gambling mode which is used by the player to place another bet or to play the game a second time using the same amount of money. Based on this string, we proved that the machine allows doubling up in a game that can be played only by betting a certain number of credits.

(b) A random number generation process

The string "special odds for total bet" indicates that a special odds payout table was called to determine the total points won by the player. The winning combination of "all fruits" is related

to random number generation, because the game element displayed in the video display cells is selected randomly from an associated random table containing the numbers and game elements [18]. The game elements described are actually typical slot machine objects (e.g., "bars", "oranges", "cherries"). When a game is played, the entire array of cells is examined. The payout table holding the "all fruits" combination of game elements is called to determine the winning combination and its associated payout value.

(c) Presence of a payout value

The presence of the "hopper empty…call attendant" string demonstrates that a player has requested a payout. At the end of play, the player decided to redeem his winnings; however a coin output error occurred. This statement was stored in memory as information related to the game played [14]. This string proves that the machine is able to redeem credits won by a player.

Our findings prove that the gaming machine described in this paper is a gambling machine. If the establishment where the machine was confiscated was unlicensed, this would constitute an illegal gambling operation.

10 Future Work

The search process could be improved by the development of software tools containing intelligent agents to perform keyword or subroutine searches and identify gambling-related terms or mechanisms within the machine memory. Current gambling machines exploit advances in system development to avoid dependence on hardware components. Advanced extraction and analysis techniques are necessary to identify gaming machines which are capable of conducting gambling activities.

Acknowledgments

Special thanks to UTM for supporting us with the facilities necessary for this research. Great thanks to the Royal Malaysian Police Force for providing us with the forensic exhibit (gambling machine). Their full cooperation assisted us in the completion of our research and is highly appreciated.

References

[1] Steven G.Lemay, Andrew M.Rodges, Robert E.Breckner, Xuedong Chen.: EPROM file system in gaming apparatus, structure of a gaming system. (2006) United States Patent No.7108605, http://www.freepatentsonline.com/7108605.html

[2] Dr. John, L., McMullan, Dr. David, C., Perrier.: Cheats At Play: The Social Organization Video Lottery Terminal Fraud. In: Gambling, Law Enforcement and Justice System Conference, Alberta Gaming Institute and University of Alberta, Edmonton, Alberta (2002)

[3] John Catsoulis.: Designing Embedded Hardware. O'Reilly, USA (2005)

[4] Marcel Breeuwsma, Martin De Jongh, Coert Klaver, Ronald van der Knijff and Mark Roeloffs.: Forensic Data Recovery from Flash Memory. Vol.1,(1). SSDDFJ (2007)

[5] Brian, D., Carrier, Joe Grand.: A Hardware-Based Memory Acquisition Procedure for Digital Investigations. Vol.1, (1). IDJE (2004)

[6] Nick, L., Petroni, Jr., AAron Walters, Timothy Fraser, William A. Arbaugh.: FATKit: A Framework for the Extraction and Analysis of Digital Forensic Data from Volatile System Memory. Digital Investigation Journal. Vol. 3, (4). (2006)

[7] Dr Benjamin Turnbull.: Forensic Investigation of the Nintendo Wii: A First Glance. Vol.2, (1). ISSN. (2008)

[8] Barry B. Brey.: The Z80 Microprocessor Hardware, Software, Programming, an Interfacing. Englewood Cliffs, N.J.: Prentice-Hall,Inc. (1988)

[9] Walter R. Siekiersi, Michael Sterling.: Random Number Generating Techniques and Gaming Equipment Employing such Techniques. (1985) United States Patent No. 4527798, http://www.freepatentsonline.com/4527798.html

[10] Dr.Elazar (Azi) Zadok, Brig. Gen. Director, D.I.F.S.: Gambling Machines Laboratory, Division of Identification and Forensic Science. Unpublished note, Investigation Department/ Israel Police Headquarters.

[11] SCI Counsel James F. Villere. Illegal Gambling. Unpublished Report, Division of Criminal Justice in the Attorney General's Department of Law and Public Safety. (1991)

[12] Memon Asim. Reverse Engineering. Unpublished note, Blekinge Institute of Technology

[13] SCI Counsel eeTools, EPROM Programmer, http://www.eetools.com/index.cfm?fuseaction=devices.do_search

[14] The National Standard Working Party.: Revision 9.0. Australian/New Zealand Gaming Machine National Standard. New Zealand: Australian and New Zealand gaming regulators. (2007)

[15] Michael J. Dietz,II, Earl D. Morris, Rolen A.Miller,: Instant, Multiple Play Gaming Ticket And Validation System. United States Patent No. 5949042, http://www.freepatentsonline.com/5949042.html

[16] Gaming Labs Certified. Standard Series. Version 2.0. Client-Server Systems. Gaming Laboratories International, Inc. (2007)

[17] Casino Gambling Terms and Definitions, http://www.bestukcasinos.co.uk/casino-terms.html

[18] John Manship, Michael Vinneau, David Ross, Nathalie Hache, Charles Maillet.: Video Gaming Machine. (1995) United States Patent No. 5393061. http://www.freepatentsonline.com/5393061.html

[19] Joseph Grand, Brian Carrier.: Method and Apparatus For Preserving Computer Memory Using Expansion Card. (2007) United States Patent No. 7181560., http://www.freepatentsonline.com/7181560.pdf

[20] Burke, Paul K. and Craiger, Philip: Xbox Forensics. Journal of Digital Forensic Practice, 1:4, 252--282 (2006)

[21] Kyung-Soo Lim, Sangjin Lee : A Methodology for Forensic Analysis of Embedded Systems. In: Second International Conference on Future Generation Communication and Networking, pp. 283—286. IEEE Computer Society (2008)

Pritheega Magalingam
Centre for Advanced Software Engineering, University Technology Malaysia
Pritheega Magalingam is a MSc.(Information Security) graduate (2009) from University Technology Malaysia. Currently, she is a research and teaching assistant in Centre for Advance Software Engineering, University Technology Malaysia. Her MSc. thesis comprises Digital Evidence Retrieval and Forensic Analysis Guideline for an Illegal Gambling Machine and she has published few papers on how to extract evidence from gambling machine and data analysis using keyword search techniques. Her current research focuses on forensic analysis tool development for evidence retrieved from electronic gaming machine. Applying artificial intelligence techniques through the development of a multiagent system that acts based on expert's knowledge of the technical domain would be her great interest.

Azizah Abdul Manaf
College Science and Technology, University Technology Malaysia
Azizah Abdul Manaf (PhD) is a Professor of Image Processing and Pattern Recognition from University Technology Malaysia (UTM). She graduated with B. Eng. (Electrical) 1980, MSc. Computer Science (1985) and PhD (Image Processing) in 1995 from UTM. Her current areas of interest and research are image processing, watermarking, steganography and computer forensics and have postgraduate students at the Masters and PhD level to assist her in these research areas. She has written numerous articles in journals and presented an extensive amount of papers at national and international conferences on her research areas. Prof. Dr. Azizah has also held management positions at the University and Faculty level such as Head of Department, Deputy Dean, Deputy Director and Academic Director pertaining to academic development as well as on training for teaching and learning methodologies at the University.

Rabiah Ahmad
Centre for Advanced Software Engineering, University Technology Malaysia
Rabiah Ahmad, Ph.D. is a senior lecturer for Information Security at Centre for Advanced Software Engineering, University Technology Malaysia (UTM). She graduated with BSc. Computer Science (UTM) in 1997, MSc. Information Security (Royal Holloway University of London, UK) 1998 and Ph.D. Information Studies (University of Sheffield, UK) 2006. Her current areas of interest and research are Threat Identification Tools for Medical Online System Using Combination Technique Genetic Algorithm and Coz Regression, Virus and Worm Analysis Using Bayesian Network and Regression Model for Healthcare System, Privacy issue in data mining and Security Architecture and Access Control. She is the author of numerous journal publications and article in the field of digital forensics, watermarking, steganography and health information management research. She presented an extensive amount of papers at national and international conferences on her research areas. Rabiah Ahmad has also held management position at the Faculty level such as Program Coordinator Master Computer Science Information Security (2004-2009), Assistant Treasurer for Malaysia Society of Cryptology Research (2009 – Present) and Academic Adviser at German Malaysian Institute (2005 – Present).

The International Journal of
FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

e-forensic
PRESS

Subscription

For subscription information, please visit the journal's web page at www.ijofcs.org

Photocopying

Single photocopies of single articles may be made for personal use as allowed by national copyright laws. Permission of the publisher and payment of a fee is required for all other photocopying, including multiple or systematic copying, copying for advertising or promotional purposes, resale, and all form of document delivery. Special rate are available for educational institutions that wish to make photocopies for non-profit educational classroom use. For more information about photocopying and permissions, please visit www.ijofcs.org

Electronic Storage or Usage

Permission of the publisher is required to store and distribute electronically any material contained in this journal, including any article or part fo an article.

Notice

No responsibility is assume by the publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwese, or from any use or operation of any methods, products, instructions or ideas contained in the material herein. Although all material is expected to conform to ethical standards, inclusion in this publication does not constitue a guarantee or endorsement of the quality or value of such product or of the claims made of it by its authors.

Publisher's Note

The opinions expressed by authors in this journal do not not necessarily reflect those of the Editor, the Editorial Board, Technical Commitee, Publishers, ABEAT, APCF, or the Brazilian Federal Police. Although every effort is made to verify the information contained in this publication, accuracy cannot be guaranteed.

Printed and bound in Brazil.

The International Journal of Forensic Computer Science

Volume 4, Number 1, 2009