Frontmatter

2  Editors
3  SIGAPP Semi-Annual Report
4  S. Shin
5  Message from the Editor
   S. Shin
   H. Haddad
   SAC 2011 Overview

Research Articles

6  A. Uhl  K. Raab
   J. Hämmerle-Uhl  Robust Watermarking in Iris Recognition
19  I. Rauf  I. Porres  Designing Level 3 Behavioral RESTful
   Web Service Interfaces
32  D. Benslimane  K. Benouaret  A. Hadjali  A Fuzzy Framework For Selecting Top-k
   Web Service Compositions
41  R. Fontugne  K. Fukuda  A Hough-transform-based Anomaly Detector
   with an Adaptive Time Interval
52  F. Rahman  S. Iqbal Ahamed
   M. Endadul Hoque  Preserving Privacy in Wireless Sensor Networks
   using Reliable Data Aggregation
Applied Computing Review

Editor in Chief
Sung Y. Shin

Designer/Technical Editor
Daniel J. Stanton

Associate Editors
Richard Chbeir
Hisham Haddad
Lorie M. Liebrock
W. Eric Wong

Editorial Board Members

Wood Alan
Davide Ancona
Papadopoulos Apostolos
Brent Auernheimer
Alessio Bechini
Giampaolo Bella
Umesh Bellur
Ateet Bhalla
Stefano Bistarelli
Olivier Boissier
Gloria Bordogna
Barrett Bryant
Alex Buckley
Artur Caetano
Luís Carriço
Rogério Carvalho
Andre Carvalho
Matteo Casadei
Jan Cederquist
Alvin Chan
Richard Chbeir
Jian-Jia Chen
Claramunt Christophe
Yvonne Coady
Luca Compagna
Arthur Wm. Conklin
Massimo Cossentino
Federico Divina
Carlos Duarte

Mario Freire
Lorenz Frohhofer
João Gama
Xiao-Shan Gao
M. Karl Goeschka
Claudio Guidi
Svein Hallsteinsen
Jason Hallstrom
Hyoil Han
A. Ramzi Haraty
Jean Hennebert
Jiman Hong
Andreas Humm
L. Robert Hutchinson
María-Eugenia Iacob
Francois Jacquet
Hasan Jamil
Robert Joan-Arinyo
Andy Keillens
Tel-Wei Kuo
Op't Martin Land
Ivan Lanese
Paola Lecca
Maria Lencastre
Va Hong Leong
Ki-Joune Li
Lorie Liebrock
Giuseppe Lipari
Shih-Hsi Liu
Rui Lopes
Marco Mamei
Eduardo Marques
Paulo Martins
Stan Matwin
Manuel Mazzara
Ronaldo Menezes
Marjan Mernik
Fabien Michel
Dominique Michelucci
Ambra Molesini
Eric Monfroy
Barry O'Sullivan
Rui Oliveira
Andrea Omicini
Fernando Osorio
Edmundo Monteiro
V. Ethan Munson
Peter Otto
Brajendra Panda
Gabriella Pasi
Manuela Pereira
G. Maria Pimentel
Antonio Cosimo Prete
Kanagasabai Rajaraman
Rajiv Ramnath
Chandan Reddy
Elliot Rich
Pedro Rodrigues
Giráldez Raúl Rojo
Alexander Romanovsky
Agostinho Rosa
David Rossi
Corrado Santoro
Rodrigo Santos
Guido Schryen
Michael Schumacher
Kumar Madhu SD
Valeria Seidita
Jean-Marc Seineur
Alessandro Sorniotti
Nigamath Sridhar
Lindsay Yan Sun
Junping Sun
Oan Chang Sun
Emiliano Tramontana
Dan Tulpan
Teresa Vazão
Mirko Viroli
Fabio Vitali
Giuseppe Vizzari
Denis Wolf
W. Eric Wong
Kokou Yetongnon
R. Osmar Zaiane
SIGAPP FY’11 Semi-Annual Report
March 2011 – August 2011
Sung Y. Shin

Mission
To further the interests of the computing professionals engaged in the development of new computing applications and to transfer the capabilities of computing technology to new problem domains.

Officers
Chair – Sung Y. Shin
South Dakota State University, USA

Vice Chair – Richard Chbeir
Bourgogne University, Dijon, France

Secretary – W. Eric Wong
University of Texas, USA

Treasurer – Lorie Liebrock
New Mexico Institute of Mining and Technology, USA

Web Master – Hisham Haddad
Kennesaw State University, USA

ACM Program Coordinator – Irene Frawley
ACM HQ

Notice to Contributing Authors

By submitting your article for distribution in this Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights.

• to publish in print on condition of acceptance by the editor
• to digitize and post your article in the electronic version of this publication
• to include the article in the ACM Digital Library
• to allow users to copy and distribute the article for noncommercial, educational or research purposes.

However, as a contributing author, you retain copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.
A Message from the Editor

With the passing of the summer, we have a new issue of ACR (Applied Computing Review). The format of this publication is settling down, and we will begin publishing quarterly electronically and in print this winter.

I still have a dream that ACR will appear in the SCI (Science Citation Index). ACR contains invited papers from world-renowned researchers and selected papers presented by prominent researchers and professionals of the Symposium on Applied Computing 2011 in Taichung, Taiwan. The selected papers have been expanded, revised, and peer-reviewed again for publishing in ACR.

We hope that ACR will serve as a platform for many new and promising ideas in the many fields of applied computing. As you know, it is strongly related to nearly every area of computer science, and we feel an obligation to serve you as best we can. The papers in this issue of ACR represent the current applied computing research trends. I thank the authors for truly contributing to the state of the art in applied computing. The officers look forward to working with the ACM SIG Governing Board to further develop SIGAPP by increasing membership and developing a new journal on applied computing. They also appreciate the opportunity to support the programs of SIGAPP since they have provided a springboard for further technical efforts and have done a great service to many technical communities.

We would like to express our sincere appreciation to the highly qualified peer reviewers that have coordinated an outstanding lineup of technical papers. We also would like to express our appreciation to the track chairs who served as the editorial board of ACR and the SIGAPP officers that served as associate editors of ACR. We would like to thank Ms. Irene Frawley as she has helped us get organized to begin publishing. I especially wish to thank Mr. Dan Stanton, technical editor of ACR, for his work on the design and layout of this new ACR, an international journal, and for his encouragement and support to publish this special issue. We also wish to thank all authors for their significant contributions. Without their hard work, this third issue of ACR would not be possible.

Next Issue
The planned release for the next issue of ACR is December 2011.
SAC Overview

The 2011 Annual edition of the ACM Symposium on Applied Computing (SAC) was held in Taichung, Taiwan, March 2011. The conference was hosted by Tunghai University. The open Call for Track Proposals and after prescreening the proposals, 40 Tracks were finally accepted for SAC 2011. The prescreening and selections were made based on the success of those Tracks in the previous editions of SAC as well as targeting new and emerging areas. The Tracks were organized into six different themes. The Symposium Proceedings and the technical presentations were focused around these themes to form a series of related track sessions.

The Call for Papers for all Tracks attracted 790 final paper submissions from 35 different countries. All submitted papers underwent the blind review process and 237 papers were finally accepted as full papers for inclusion in the Conference Proceedings and presentation during the Symposium. The final acceptance rate for SAC 2011 is 30% among all tracks. In addition, 59 papers that received high review scores were invited as short papers for presentation during the Poster Program. The posters were presented over two sessions on Wednesday. Monday Tutorials Program offered 5 tutorials covering verity of topics and attracting over 80 attendees. The three-day Technical Program included 243 presentations from forty tracks covering a wide range of topics on applied computing and emerging technologies. For more details, please visit SAC 2011 website at http://www.acm.org/conferences/sac/sac2011/.

The success of the technical program was made possible through the hard work of many people from the scientific community who have volunteered and committed many hours to make it a success. Much credit goes to all the Track Chairs and their Program Committees. On behalf of SAC 2011 Organizing Committee and Steering Committee, we congratulate all of the authors for having their papers accepted in their respective Tracks. We also wish to thank all of those who made this year’s technical program a successful one, including the speakers, track chairs, reviewers, program committee members, session chairs, presenters, and attendees. The social program was organized and made possible by the local organizing committee. Credit goes to the local sponsors and members of the local committee for their hard work to plan and execute all aspects of the conference, including the daily coffee breaks and lunches; the receptions and their entertainment, the banquet and its entertainment; the transportation to and from the conference site; the meeting rooms and AV support; and the support for the technical meetings. Special thanks go to the student volunteers for their unprecedented help and support.

The preparation for the 2012 edition is underway. SAC 2012 will be held in Riva del Garda, Trento, at the Convention Center (Congress). It is hosted by the Centre for Computational and Systems Biology (COSBI): The Microsoft Research - University of Trento, Italy. The local organizing committee is lead by Dr. Paola Lecca and Professor Mirtis Conci, from COSBI. A member of SAC Steering Committee, Dr. Sascha Ossowski, from the University Rey Juan Carlos, Madrid, Spain, will serve as the Conference Chair. Local sponsors include Provincia Autonoma Di Trento and Riva del Garda Congress. The conference dates are set for March 25 – 29, 2012. We hope you consider SAC 2012 for your next submission and hope to see you there next year. The complete organizing committee and other information can be found on SAC 2012 website at http://www.acm.org/conferences/sac/sac2012/.
Robust Watermarking in Iris Recognition: Application Scenarios and Impact on Recognition Performance

Jutta Hämmerle-Uhl, Karl Raab, Andreas Uhl
Multimedia Signal Processing and Security Lab (WaveLab)
Department of Computer Sciences
Salzburg University, Austria
uhl@cosy.sbg.ac.at

ABSTRACT
Watermarking has been suggested as a means to improve security of biometric systems. We discuss application scenarios for resolving various security issues in this context. As embedding watermarks changes biometric data, degraded recognition performance using such data may be expected. We experimentally investigate the impact of applying a set of blind robust watermarking schemes on the recognition performance of two iris recognition algorithms. We find that different watermarking schemes result in a very different amount of impact rendering the choice of a particular watermarking scheme an important issue to be considered in the investigated context.

Categories and Subject Descriptors

General Terms
Security, Verification

Keywords
Watermarking, Biometrics, Iris Recognition

1. INTRODUCTION
Biometric recognition applications become more and more popular. Many institutions, governmental agencies and companies want to rely on this upcoming technology to secure their environment because standard authentication methods like PINs, passwords, smart-cards etc. have many disadvantages. Possession and knowledge based authentication techniques are prone to human errors since the former can be lost and the latter can be forgotten. Moreover, these technologies can be applied without actually guaranteeing a specific human presence. Biometric authentication systems can resolve most of these issues, since a biometric feature belongs only to one person and cannot be lost or forgotten. But eventually, biometric features can be stolen or adopted and there exist various other ways to circumvent the integrity of a biometric authentication system (see e.g., a corresponding collection of security issues compiled by the UK Government Biometrics Working Group\(^1\). Recent work systematically identifies security threats against biometric systems and possible countermeasures (among them watermarking) \([37]\) and discusses man-in-the-middle attacks and BioPhishing against a web-based biometric authentication system \([50]\). In their classical paper \([35]\) Ratha et al. identified and described several stages at which a biometric system may be attacked by an intruder or imposter:

1. Fake the biometric trait of a genuine user at the sensor (e.g., fake finger or printed face image)
2. The transmission between sensor and feature extractor may be intercepted and resubmitted by changed or replayed data
3. Override the feature extractor to produce predefined feature sets
4. Intercept and replace the extracted feature sets by a synthetic or spoofed one
5. Override the matcher to always produce high matching scores
6. Modify, replace, remove stored or add new templates at the database
7. Intercept the communication channel between template database and matcher
8. Override the final decision

Spoofing a physiological biometric feature at the sensor site can be seen as counterpart to exposing a password. If a fraudulent duplicate or sample data previously acquired is accepted by the sensor, breaking a biometric system is at least as easy as spying out a password. Having acquired these “raw image data”, a dedicated attack against the targeted person would be facilitated in case no further security mechanisms are employed. The acquired image data could be presented to the sensor (item 1.) or could be inserted into the transmission of data between sensor and feature extractor (item 2.). Since also the feature set extraction is straightforward given the raw image data, the computed

\(^1\)http://www.cesg.gov.uk/policy_technologies/biometrics/media/biometricsecurityconcerns.pdf
Watermarking (WM) has been suggested as a means to resolve some of these security breaches in the last years. However, several issues have not been resolved or investigated thoroughly in this context. While the aim of steganographic approaches in the biometric context is rather obvious (i.e., to covertly communicate biometric data), many of the publications about WM schemes to protect biometric image data do not address the question what the proposed method is actually applicable for (e.g., in which stage of the biometric recognition process WM should be applied, which data should be embedded etc.) and lack of a definition of concrete use cases, motivation, target scenario, possible attacks, and alternative cryptographic techniques. Often, also an in-depth discussion of required WM properties is missing for the scenarios described in a vague manner.

A second issue not investigated in sufficient detail is the possible impact of applied WM methods on the recognition accuracy of the biometric systems that should be protected with this technology (this question of course only applies to WM scenarios, where the host data actually are biometric sample data which are used in biometric recognition and have been modified by the WM process). The aim of this paper is to address the issue of possible recognition performance degradation when using watermarked biometric data and to advance the knowledge concerning the interference of watermarking and biometric recognition. In particular, we investigate whether certain WM techniques are more suited then others and we also address the influence of WM parameters (e.g., embedding strength, capacity) and WM properties (e.g., embedding domain, embedding frequency band, and embedding technique) onto possible recognition accuracy degradation.

Section 2 provides a compact review of literature devoted to watermarking in biometrics with focus on application scenarios and intended use of the technology. In Section 3 we describe large scale experiments, where we apply 10 different WM schemes to iris sample data with different settings and study the impact on the recognition performance of two distinct iris recognition schemes. Section 4 discusses the results and concludes the paper.

2. WATERMARKING IN BIOMETRIC SYSTEMS
2.1 State of the art

Watermarks have been suggested to be used in the context of many biometric modalities, including speech, signatures, handprints, iris, and many more. Most publications are seen for fingerprint data. One of the first ideas to somehow combine biometric technologies and watermarking is “biometric watermarking”. The aim of watermarking in this approach is not to improve any biometric system, but to employ biometric templates as “message” to be embedded in classical robust watermarking applications like copyright protection in order to enable biometric recognition after the extraction of the watermark (WM). As a consequence, the WM has to be capable of carrying the template data (capacity requirement) and should not be perceived. The robust WM has to resist against unintentional and malicious cover data manipulations.

Vielhauer et al. [43] introduce the general concept and notion of biometric watermarks, also discussed in [1]. One of the most interesting applications in the context is the “secure digital camera” [4], where an iris template of the photographer is embedded into digital images. Canon filed a corresponding patent recently (US Patent Application No. 2008/0025574). A similar idea also addressing image integrity is proposed in [16]. Low et al. [27] suggest to embed offline signatures into digital images for copyright protection.

In order to motivate the use of watermarking in the biometric context with the aim of improving security, Jain et al. [19] suggest that if only traditional cryptographic techniques are used for the protection of biometric data, the data has to be decrypted somewhere along the way and therefore after decryption, security for the data is not maintained anymore – here watermarking comes in as a “second line of defence” similar to the DRM scenario since a watermark is still present after decryption. In this manner, information carried by the watermark can still be retrieved even if cryptographic tools have already been defeated.

There has been a lot of work done during the last years proposing watermarking techniques to enhance security of biometric systems in some way. Dong et al. [12] try to give a systematic view of the situation in the case of iris recognition by distinguishing whether biometric template data are embedded into some host data (“template embedding”), or biometric sample data is watermarked by embedding some data into them (“sample watermarking”). In the latter case, they distinguish between robust embedding techniques for database ownership protection and fragile techniques for sample tampering detection.

The impact of watermarking on the recognition performance of biometric systems has been investigated most thoroughly in the context of iris recognition also. While Dong et al. [12] do not report on performance degradations when investigating a single watermark embedding algorithm and one iris recognition technique only, Hämmerle et al. [15] find partially significant reductions in recognition accuracy when assessing two iris recognition schemes and a couple of robust watermarking algorithms. For fingerprint recognition, wa-
termarking techniques aiming at negligible impact on recognition performance have been designed explicitly. This is achieved for example by applying two blind robust spatial watermarking methods embedding a character bit string either sparing out fingerprint feature regions (i.e., close to minutiae data) or by maintaining the ridge gradient orientations [39, 7]. This approach has been followed by many other techniques (e.g., [10, 30, 21]). On the other hand, recent work by Zebbeche et al. [48, 49] proposes two robust WM schemes for fingerprint images where WM data is embedded into the ridge area (region of interest RoI) only. The aim is to increase robustness of WM due to the concentration onto the RoI, while some impact on recognition performance may be expected by using this idea.

Hong et al. [18] discuss the application of robust watermarking and symmetric encryption techniques for the exchange of compressed biometric sample data, where they also investigate the impact on accuracy of a fingerprint recognition scheme. Additionally, energy consumption of different variants in a distributed authentication scenario with mobile sensors is investigated.

2.2 Application scenarios

A first application case for robust WMs is to prevent the use of sniffed sample data to fool the sensor in order to complement or replace liveness detection techniques. During data acquisition, the sensor (i.e., a camera) embeds a WM into the acquired sample image before transmitting it to the feature extraction module. In case an intruder interferes the communication channel, sniffs the image data and presents the fake biometric trait (i.e., the image) to the sensor, it can detect the WM, will deduce non-liveness and will refuse to process the data further. This idea may also be applied to biometric databases where e.g., Bartlow et al. [3] propose a framework that encodes voice feature descriptors in raw iris images stored in a database.

An entirely different aim of applying robust embedding techniques to sample data is proposed in [38]. Here, the embedded signature is used as an additional security token like an additional password, which basically leads to a two-factor authentication system, based on biometrics and the additionally embedded secret data.

A steganographic approach is to transmit biometric data (i.e., template data) hidden into some arbitrary carrier / host data or biometric samples of different biometric modalities. The idea is to conceal the fact that biometric data transfer takes place, e.g., Jain et al. [20] propose to embed fingerprint minutiae into an arbitrary host image while Khan et al. [22] suggest to embed fingerprint templates into audio signals.

Questions of sensor and sample authentication using watermarks have also been discussed. During data acquisition, the sensor (i.e., a camera) embeds a watermark into the acquired sample image before transmitting it to the feature extraction module. The feature extraction module only proceeds with its tasks if the WM can be extracted correctly. For example, fragile watermarking has been suggested to serve that purpose either embedding image-independent [47] or image-dependent data as WM [44]. Ratha et al. [36] propose to embed a response to a authentication challenge sent out by a server into a WSQ compressed fingerprint image in fragile manner in order to authenticate the sensor capturing the fingerprint image.

A significant amount of work has also been published in the area of using WMs to enable a multibiometric approach by embedding a biometric template into a biometric sample of different biometric modalities. There are two variants: First, there are two different sensors acquiring two biometric traits. Since for one modality template data is embedded, these data need to be generated at the sensor site which makes this approach somewhat unrealistic, at least for low power sensor devices. In addition to that, besides the increased recognition performance of multimodal systems in general there is no further specific gain in security. The second variant is to store the template on a smart-card which has to be submitted by the holder at the access control site. The smart-card embeds the template into the host sample data. This in fact represents a two-factor authentication system which increases security by introducing an additional token-based scheme and also leads to higher recognition accuracy as compared to a single biometric modality.

Hoang et al. [17] embed fingerprint minutiae in facial images (with fragile watermarks), while Jain et al. [21] embed face data into fingerprint images using a technique classified as being robust. Chung et al. [10, 30] use the same watermarking technique as well to embed fingerprint templates into facial images and vice versa, and compare the recognition performance of the resulting systems. They also use this embedding technique as the fragile part of a dual watermarking approach [30, 23] so that doubts remain about the actual robustness properties of the scheme. Vatsa et al. employ robust embedding techniques: in [41], they embed voice features in colour facial images, the same group [31, 40, 42] propose to embed facial template data (and additional text data in the first work) into fingerprint sample data using a robust (multiple) watermarking approach. Park et al. [33] suggest to use robust embedding of iris templates into face image data to enable various functionalities, Kim et al. [24] propose a blind and robust spread spectrum watermarking technique for embedding face template data into fingerprint samples.

Thus, we have identified three different application scenarios, where robust WM have been proposed to be employed in order to embed information into sample data: First, embedding of sensor ID-information to prevent replay attacks of sniffed sample data. Second, embedding of signature information of a two-factor authentication system. And third (which also represents the most important application case), embedding of template data into sample data to enable a multibiometric system. In all three application scenarios, a degradation of biometric recognition performance caused by the embedded watermark is highly undesired.

3. EXPERIMENTAL INVESTIGATION

As we have seen, iris related data in the context of watermarking is not yet covered to a significant extent in literature (since most techniques focus on fingerprint images) – only [3] embeds data (i.e., voice descriptors) into iris sample images while [32] embeds iris template data into face images. Fig.
1 illustrates our experimental scenario. The iris scan is acquired at the sensor site and the raw image is watermarked and transmitted over a network. At the receiver, WM detection is performed and finally, iris feature extraction and matching is done. Note that the considered experimental scenario does not target a single specific application scenario as outlined before. The experiments conducted serve the purpose to investigate the impact on matching performance caused by embedded watermarks and do not at all give information about the security of an application scenario using robust WM embedding.

**Figure 1: Considered application scenario**

Since the WM embedding stage introduces noise-like patterns into the biometric sample data, an impact on recognition performance may be expected naturally.

### 3.1 Experimental Settings

#### 3.1.1 Watermarking Algorithms

Since there is no realistic biometric application scenario imaginable where there is the unmarked original image available to detect the WM in the marked image, only blind WM techniques are applicable in this context. We consider a variety of blind robust WM techniques in different flavours in order to study potentially different results when distinctive embedding techniques are used. We have used some algorithms from the “Watermarking Toolbox” developed by Peter Meerwald for WM embedding and detection [29], additionally more recent wavelet-based schemes have been implemented as well (i.e. Barni, Cao, Chen, Pla, and Wu, see below). Using the WM techniques as listed subsequently, we have representatives for a variety of embedding domains (spatial domain, DCT domain, wavelet approximation domain, wavelet detail subbands), representatives for different embedding strategies (additive spread spectrum vs. quantisation-based WM), as well as representatives for different types of WM (binary vs. Gaussian distributed values).

### Spatial and DCT based algorithms

**Bruyndonckx (shorthanded Bruyn)** This algorithm operates on 8x8 blocks with modifications on the luminance values where each block is able to obtain one bit of information. A binary sequence is used as watermark [6].

**Koch** Uses a random sequence of concrete image positions. At this positions the DCT coefficients of 8x8 blocks are used for embedding imposing a strict ordering. The watermark is a binary sequence [25].

### Wavelet-based algorithms

**Barni** 4-level decomposition. Additive embedding in the 3 finest detail subbands with visual masking. The watermark is a pseudo random binary sequence [2].

**Cao** Uses a redundant wavelet transform and embeds in the 3 finest detail subbands. Additive embedding by creating a significance mask. A Gaussian distributed random sequence is used as watermark [8].

**Chen** Embedding is done in the approximation subband depending on the watermark length via a bit selection algorithm. The watermark is a black/white image (i.e., a binary sequence) [9].

**Dugad** 3-level decomposition. Uses a Gaussian-distributed watermark sequence. Additive embedding is applied only to a few significant coefficients using an image sized watermark in the detail subbands [13].

**Kundur** 3-level decomposition. Uses a binary value sequence. Locations for embedding are pseudo-randomly selected in the detail subbands. A triple of coefficients at different subbands within the same spatial position is selected. The middle coefficient is quantised [26].

**Pla** Additive proportional embedding in significant trees of coefficients in the detail subbands by visual modelling. The watermark is a Gaussian distributed random sequence [34].

**Wu** 3-level decomposition. The quantisation based watermark is embedded in trees of coefficients. A random binary sequence is used as watermark [45].

**Xie** Embedding only in the approximation subband. Selecting the middle of three coefficients by a sliding window. The middle coefficient is then quantised. The watermark is a binary sequence [46].

The implementation of the Watermarking Toolbox (all algorithms but Chen, Barni, Cao, Pla and Wu) allows the free selection of the signature length. We chose either a “normal” standard signature length (128 bits for binary marks or 1000 numbers for Gaussian-distributed marks), or a “long” signature length (1024 bits for binary marks or 32000 numbers for Gaussian-distributed marks) to check the signature length influence. For “Xie” the length is limited by the size of the approximation subband (here 80 bits), therefore for this algorithm only the “normal” signature length is available. The same applies to the other algorithms (i.e., those not contained in the Watermarking toolbox), where the signature length is dependent on the image size.

Note that the question of WM capacity can be an important one in the biometric context – e.g., if we intend to embed iris template data, the Daugman schemes requires 2048 bits to be embedded. Embedding other types of information like a camera ID to authenticate an admissible biometric sensor requires significantly lower capacity.

#### 3.1.2 Iris Recognition Algorithms

Most iris recognition methods follow a common structure [5], close to the well known and commercially most successful approach by Daugman [11]. After image acquisition, in a first step the iris texture is localised and extracted. From this texture, discriminative features are derived, which then can be used for comparison.

---

3http://www.cosy.sbg.ac.at/~pmeerw/\Watermarking/source/
In our approach (following e.g., Ma et al. [28]) we assume the texture is the area between the two almost concentric circles of the pupil and the outer iris. These two circles are found by contrast adjustment, followed by Canny edge detection and Hough transformation. After the circles are detected, unwrapping along polar coordinates is done to obtain a rectangular texture of the iris. In our case, we always resample the texture to a size of 512x64 pixels. The texture is divided into N stripes to obtain N one-dimensional signals, each one averaged from the pixels of M adjacent rows. We used N = 10 and M = 5 for our 512x64 pixel textures (only the 50 rows close to the pupil are used from the 64 rows, as suggested in [28]).

Working now only on these data, we employ two distinct techniques to extract a bit-code. The first approach (denoted as “KLV”) has been proposed by Ma et al. [28]. A dyadic wavelet transform is performed on each of the resulting 10 signals, and two fixed subbands are selected from each transform. This leads to a total of 20 subbands. In each subband we then locate all local minima and maxima above some threshold, and write a bitcode alternating between 0 and 1 at each extreme point. Using 512 bits per signal, the final code is then 512x20 bit. The second approach (denoted as “LG”) is similar to Libor Masek’s Matlab implementation of a 1-D version of the Daugman iris recognition algorithm. A row-wise convolution with a complex Log-Gabor filter is performed on the texture pixels. The phase angle of the resulting complex value for each pixel is discretized into 2 bits. The 2 bit of phase information are used to generate a binary code, which therefore is again 512x20 bit. For both techniques, matching is done by computing the Hamming distance between two binary codes at several shifted positions retaining the minimal value as distance between two irises.

The tests actually conducted are based on the CASIA-IrisV3 image database. Out of this database we use the left-eyes of the “Interval”-samples only. We have computed about 800,000 intra-class comparisons on the basis of 894 images of 143 persons (6.25 images per person on average) and a corresponding number of inter-class comparisons. For our tests, we trim the watermark embedding strength to achieve an average PSNR after embedding of about 42dB and 30dB in order to consider two different application cases. While with 42dB the WM is embedded without causing perceptual degradations (high quality but low robustness), 30dB represents an application case with high robustness but already perceivable WM artefacts. The signature is changed in every embedding process to blur its influence. So the result for one watermarking algorithm consists of 33760 different signatures (894 images × 2 embedding strengths × 2 recognition algorithms × 10 runs).

Due to this enormous computational demand we were able to generate results for 10 runs only. The minimal and maximal values are given and can give an estimation of the variation, though.

### 3.2 Experimental Results

The following Receiver Operating Characteristics (ROC) curves are used to illustrate the iris recognition performance in case the data used in recognition has been watermarked. In the case of watermarking being applied, one of the two iris images involved in the matching process has been watermarked. Consequently we cover the following two application cases: either the sample image has been watermarked or the templates have been subject to watermarking during enrollment, but not both. In the graphs, the False-Match-Rate (FMR) is plotted against the X-axis while False-Non-Match-Rate (FNMR) is plotted against the Y-axis.

Inside the plots, the “no watermark” curve shows the performance of the iris recognition algorithm without watermarking being applied to the the iris images. In case watermarking being applied, the 42dB and 30dB curves are the average results, calculated over 10 test runs each. The min/max bars indicate the minimum and the maximum values to estimate variations due to different watermarking signatures applied. “normal” and “long” refers to the signature lengths, respectively. On each ROC curve the equal-error-rate (EER) is depicted as a single symbol (dot, square, star).

We first discuss the results of the KLV iris recognition algorithm. In Fig. 2 we compare the ROC curves of the Barni and Chen watermarking methods, respectively. Recall that the Barni technique embeds the WM into the wavelet detail subbands whereas Chen uses the approximation subband for embedding. The Barni WM approach does not degrade recognition results too much, however, we notice a slight reduction of recognition accuracy across the entire range of investigated matching rates, with higher impact when embedding with 30 dB as compared to embedding with 42 dB (as it is expected of course).

The situation is different with Chen WM. The ROC curve when embedding with 42 dB is slightly worse as compared to that of Barni WM with 30 dB (EER 0.023 vs. 0.022), as are the variations between best and worst cases as depicted in the plot. When embedding with 30 dB, ROC behaviour is significantly worse (EER 0.05) as are the fluctuations between different runs of the test.

Fig. 3 displays a similar comparison. Dugad WM selects the embedding sites in the wavelet detail subbands while Xie WM embedding takes place in the approximation subband. The results of Dugad WM is almost identical to Barni WM, whereas Xie WM again significantly degrades the recognition behaviour in a similar manner as Chen WM does (in the Xie case, at 30 dB the results are better, but this is hardly noticeable).

These results indicate, that indeed using the approximation subband for WM embedding has a significantly higher impact as compared to detail subband embedding in case of wavelet based WM. This is plausible, since the information contained in the approximation subband represents the basis for a correct representation of the iris texture. In particular, the approximation subband contains a low-resolution and averaged version of the iris texture with local details removed. In case of damage of this information by adding WM information, the incorrect data spreads out into the reconstruction of the texture, generating artificial details in

---

many areas and thereby inhibiting good matching results. In contrast, modification of detail subband information affects the texture only locally and does not degrade results that much.

Fig. 4 shows the ROC results for the spatial Bruyn WM algorithm and compares embedding a normal vs. a long WM. For embedding with 42 dB, the results are competitive to the behaviour of Barni and Dugad WM for both signature lengths considered. When embedding with 30 dB we notice very different results comparing normal and long signatures. While for normal signature length recognition accuracy at 30 dB is almost equivalent to Barni and Dugad WM, the long signature case behaves similar to Chen or Xie type embedding.

The case of the block-DCT based Koch WM is covered in Fig. 5. We find a behaviour very similar to the Bruyn WM scheme discussed before, however, in case of normal signature length and 42 dB embedding we obtain the best results of all WM schemes investigated (least impact on recognition accuracy). Also the other results are better compared to Bruyn, but the overall trend is identical (significant impact of the long signature at 30 dB embedding).

The results of the Bruyn and Koch WM schemes indicate that no matter which embedding domain is selected, we notice an impact on recognition results with lower or higher extent. Both schemes operate in a block based manner which means that for longer signatures, more blocks have to be manipulated for embedding. At 30 dB embedding, this results in a significant impact on the recognition results. As we can observe in Table 1, for wavelet based WM schemes using detail subband embedding, no additional recognition degradation caused by long signatures can be observed. In these techniques, longer signatures only affect single coefficients (as opposed to entire pixel blocks) which results in lower on recognition results. When comparing Wavelet-, DCT-, and Spatial-domain-based embedding we overall find localized embedding strategies showing the least impact on recognition performance (since here only single bits in the templates are affected), while the errors caused by the approximation subband based embedding cause more burst-type errors causing much more significant impact on recognition.

The results for the LG iris recognition algorithm confirm the behaviour observed so far. First, we provide detailed ROC results again for the comparison between wavelet-based approximation and detail subband embedding strategies. In
Fig. 2, we notice that the overall results for the LG scheme are worse compared to KLV (also without any WM applied). The results of Chen WM is even more degraded as compared to Barni WM as we have observed for the KLV approach. Here, at 30 dB WM embedding strength, the scheme with WM employed does not lead to usable results any more. Moreover, Chen WM at 42 dB embedding strength is even worse compared to the Barni scheme at 30 dB embedding.

Similar behaviour may be observed when comparing Dugad and Xie WM as shown in Fig. 7. Xie WM at 42 dB is clearly inferior to Dugad WM at 30 dB which clearly confirms the observations made in case of KLV iris recognition. Also for Xie at 30 dB the results are hardly useful any more.

Contrasting to all the other WM techniques which embed in the detail subbands (Barni, Cao, Dugad, Pla), the Wu embedding approach suffers significantly from embedding with higher strength as shown in Fig. 8. This behaviour is also confirmed for the KLV algorithm in the subfigure below. As we shall see in Table 1, this effect can also be observed for a second algorithm which is based on detail coefficient embedding, i.e., the Kundur approach.

Turning to spatial and DCT base embedding schemes we again focus on the potential effect of longer signatures. For the Bruyn embedding algorithm, the KLV results are confirmed in that we observe significant impact for stronger embedding with high signature length only (Fig. 9). At 42 dB embedding, the results for normal and long signatures are almost identical.

For the Koch WM approach as shown in Fig. 10 we again find almost identical ROC for normal and long signatures while the recognition accuracy at 42 dB is the best of all schemes considered. Again, the impact of 30 dB embedding is significantly higher for long signatures as compared to "normal" ones.

Since it is not sensible to display the ROC curves of all WM algorithms with all possible parameters we have selected the most interesting results for the displayed details so far. In order to provide a (more crude) overall view, Table 1 provides the KLV and LG EER of all watermarking schemes investigated (where "bruynL" denotes the employment of a long signature as opposed to "bruyn" with a normal signature length). We notice that the actual iris recognition scheme employed does not at all make any difference in the results, the effect of the WM schemes is entirely identical on KLV and LG.
Figure 6: ROC of LG iris recognition using watermarked authentication samples, normal signature length.

The WM schemes with lowest impact on recognition results at 42 dB and 30 dB embedding are Koch, Barni, and Pla. Considering the four schemes which enable long signatures to be embedded, Dugad WM shows the best behaviour with almost no difference between normal and long signature.

Apart from the observation made so far with respect to wavelet-based approximation and detail subband embedding we notice an additional issue: The Kundur and Wu WM schemes also exhibit a higher impact on recognition results at 30 dB embedding, although these techniques apply embedding in the wavelet detail subbands (as already observed for Wu in Fig. 8). However, there is an important difference as compared to the other detail subband embedding schemes (Barni, Cao, Dugad, Pla) concerning the underlying WM approach: while the latter four algorithms use an additive embedding rule, Kundur and Wu apply quantisation based embedding. Therefore, one might conjecture that quantisation-based embedding results in higher impact on recognition results as compared to additive embedding.

4. CONCLUSION

In this paper we have investigated if the employment of robust WM schemes eventually degrades the recognition per-

Table 1: EER for KLV and LG iris recognition algorithms under watermarking (EER is 0.014 and 0.028, respectively, without watermarking being applied).
performance of iris recognition systems as it might be expected due to the changes performed on the biometric data in case of WM being applied. As to be expected, watermarking actually affects the iris recognition performance. For the two iris recognition algorithms considered, the observed trends are almost identical.

We have found significant differences among the different WM algorithms with respect to impact on recognition results, ranging from almost no degradation to making the results almost useless. This implies that when designing a WM scheme for employment in a biometric context, it is important to consider those issues. While WM embedding at 42 dB (high image quality but low robustness) results in minor impact on recognition performance, embedding with 30 dB (lower image quality but higher robustness) causes significant recognition degradations for some algorithms. It is a question of the specific application scenario, if high WM robustness is required for security or not.

While the results of this study are far from being conclusive with respect to the question which WM schemes are well suited for this application area and which are not, we have observed certain trends as follows. WM schemes exhibiting low impact on the recognition results can be designed in each embedding domain (e.g., spatial, DCT, and wavelet). Wavelet-based WM embedding schemes tend to degrade recognition results in case embedding is done in the approximation subband and to a lesser extent in case quantisation based embedding is applied (even in case the embedding locations are situated in the detail subbands). We have seen that some embedding strategies are fairly sensitive if long signatures are used: this is observed for block-based embedding (in the spatial and DCT domains, respectively) and again for quantisation-based embedding.

It has to be noted that we have not considered the different robustness properties of the WM schemes under investigation (or other properties of relevance in the biometric context like computational demand etc.). Here, we have applied a normalisation with respect to a fixed image degradation caused by embedding (i.e., 42 dB and 30 dB), an other option would be to normalise with respect to e.g., a fixed robustness against a specific attack (resulting in a different image degradation for each WM scheme).

An entirely different approach is to rely on reversible WM schemes [14] which enable to reconstruct the original signal after WM extraction. This property (which is important
e.g., in applying WM to medical imagery) fits perfectly into the biometric scenario since it enables recognition with entirely unaffected sample data.

**Acknowledgements**

This work has been partially supported by the Austrian Science Fund, TRP project no. 554. We thank Christian Koidl and Peter Meerwald for their help in using their watermarking software and other WaveLab members for the inspiring discussions on the topic. Additional thanks go to the ACM SAC 2010 Applied Biometrics Track Chairs for their continuous support of this work.

5. REFERENCES


[17] T. Hoang, D. Tran, and D. Sharma. Remote


About the authors:

Jutta Hämmerle-Uhl works as part-time Post-Doc at the Computer Sciences Department at Salzburg University (Austria). She also works with her own company Horus IT GmbH in the area of information systems.

Her research interests include database technology, biometrics, and media security.

Karl Raab wrote his Masters thesis on watermarking in biometric systems and for scalable media types. Currently, he works as a systems engineer.

Andreas Uhl is an associate professor at the Computer Sciences Department of Salzburg University (Austria) where he leads the Multimedia Signal Processing and Security Lab. He is also a lecturer at the Carinthia Tech Institute and the Salzburg University of Applied Sciences.

His research interests include image and video processing, wavelets, media security, biometrics, compression techniques, and number theoretical numerics.
Designing Level 3 Behavioral RESTful Web Service Interfaces

Irum Rauf and Ivan Porres
Åbo Akademi University, Dept. of Information Technologies, Turku, Finland
irum.rauf@abo.fi, ivan.porres@abo.fi

ABSTRACT
A web service interface contains information about the names of the operations that can be invoked on the service and the input and output parameters of these operations. This information is not enough to facilitate service developer and consumer in understanding the behavior of the service. In the context of RESTful web services, the requirements of RESTful interface should also be met that are not fulfilled by just advertising the allowed operations on the resources. In addition, RESTful services take hypermedia as an engine of application states. Such services are defined to be at level 3 of Richardson Maturity Model (RMM). In this paper, we present an approach to model the structural and behavioral interface of a RESTful web service using UML class and UML protocol diagrams. These models lead to RESTful interfaces that conform to level 3 of RMM and describe the behavior of operations in terms of preconditions and post-conditions. These models facilitate the authentication mechanism and provide clear mapping to HTTP requests and responses. The generated contracts of methods can be published in an extended version of the WADL language and also used for documentation, stub generation, testing and monitoring purposes.

Keywords
REST, contract, UML, protocol state machine, WADL, behavioral interface

1. INTRODUCTION
A web service provides a programmatic interface to a computer system offered over a network using standard web protocols. Web services are consumed by other programs and services. This is unlike the document web or an interactive web application that is designed for human users who search for a particular information, receive it in a format that they understand and then use it manually as input to another software or machine for further processing.

The Representational State Transfer (REST) architectural style has become a popular approach to design web services. It emphasizes on generality of interfaces and scalability of component interactions. A web service that follows the REST style is often called a RESTful web service.

Many RESTful web services present simple interfaces to create, retrieve, update and delete information from a database (also called CRUD interfaces). However, REST is not limited to simple CRUD interfaces. It is possible to create web services with a complex application state that still follow the REST architectural style. In these cases, it is important to create and publish behavioral service interfaces so other developers can understand how to use a service correctly.

The REST services that provide beyond CRUD behavior have rich applications states that need to be preserved using a stateless protocol. The usage of a stateless protocol like HTTP as an application protocol facilitates the creation of scalable applications since each request is treated independent to another. Thus creating RESTful web services with rich applications states is an interesting design challenge. RESTful services have their own design requirements. Below, we provide brief background of these requirements and additional features that need to be catered for the construction of behavioral RESTful web services interfaces and their validation.

1.1 REST Interfaces
The REST architectural style is defined by four attributes. In the context of a web service these attributes are:

- **Addressability:** The REST style requires that any important piece of information related to a service should be exposed as a resource and each resource should be addressable via a URI.
- **Connectedness:** This requires that resource representation contains links to other resources.
- **Uniform Interface:** All resources are manipulated using the standard HTTP methods. The HTTP GET, POST, PUT and DELETE are used to retrieve information from a resource or change its state.
- **Statelessness:** There is no hidden session or state information. Besides, the effects of the POST, PUT and DELETE operations should be observable in the affected resources.

Any RESTful web service should comply with these four attributes. Therefore these attributes become requirements over the design of our web service interface.


1.2 Richardson Maturity Model

HTTP is a stateless protocol and each HTTP request is treated independently of any previous request. However, interactive web applications often require that the state of the application is preserved such that a new request is in relation to the previous request. Such applications take client through a sequence of HTTP requests in a particular order to fulfill a task. RESTful web services take HTTP as an application protocol and require that same set of methods (HTTP verbs) are invoked on each resource offering uniform interface. But in order to create RESTful web services with stateful behavior, there should be a provision to carry the state of the application from one independent HTTP request to another. The best way to do this is to provide links in the representation of the resources [20]. These links can contain information from the server on what further links should be addressed so that sequence of method invocation is maintained and also state of the application is preserved. These URIs can contain state information either encoded in itself or may point to the data store where the state information is saved. Thus using stateless HTTP protocol, stateful applications can be constructed in this manner. The objective behind this is to create stateful scalable REST web services.

However, not all web services are created in a manner that employ the potential of web, not even many services that claim to be RESTful. In order to identify maturity of web services in terms of RESTful behavior, Richardson Maturity Model (RMM) [20] presents a classification of web services to quantify maturity of web services. The model provides three level of maturity levels of services based on their support for URIs, HTTP and hypermedia.

- Level 0 services are the basic level services that use a single URI and a single HTTP method
- Level 1 services employ many addressable resources but only single HTTP verb
- Level 2 services use several URI addressable resources and support several HTTP verbs on these exposed resources.
- Level 3 services, in addition to the URI addressable resources and support of several HTTP verbs, contain URI links to other resources in their representations that might be of interest to the consumer of the service.

In this classification of web services, the Level 3 services are said to be most web-aware since they take hypermedia as the engine of application states. We require that web services created using our design approach are purely RESTful with maturity of level 3 according to RMM.

1.3 Method Contracts

The interface of a web service advertises the operations that can be invoked on it. A web service developer looking for a particular service finds the service over the web and integrates it with other services by invoking the advertised operations and providing it the required parameters. These operations may require a certain order of invocation or there may be special conditions under which they can be invoked.

This information, together with the expected effect of an operation forms part of the behavioral interface of a service.

We require a design approach that preserves the sequence of method invocation and is rich in behavioral information specifying the condition under which methods can be invoked and what are the expected results. This information can then be extracted from the models and asserted in the interface description language to constraint the user of the service to invoke the method under right conditions and also constraint the implementation to provide the functionality that is expected from it.

The behavioral interface can also be used to test a service implementation and for service discovery. More advanced scenarios, such as automatic service discovery and service repositories rely in formal descriptions of services.

1.4 Authentication

A secure web involves the concept of authenticating the parties involved in an interaction. Authorization determines if the user has the right to get a response for a given request. Different parties may be involved in creation of a web service. These parties may invoke different methods resulting in different application states. In addition, these HTTP requests may involve request parameters that are passed along with the HTTP requests in accordance to statelessness feature of RESTful interface (having no information in session or hidden variables). Our fourth requirement is to provide enough information to the developer through models so that it facilitates the creation of secure web services and a direct mapping to HTTP requests and responses.

We address all these aspects to create secure and pure RESTful web service from the design phase. By pure RESTful web services we mean RESTful services that take hypermedia as an engine of application states. The objective of our work is twofold. First, to provide a modeling approach that ensure that the designed interface follows the REST style. Second, to provide a way to describe behavioral services interfaces that specifies how to use a web service correctly and what are the expected results of using a web service.

We claim that our models are rich in behavioral specifications as we address different aspects of RESTful web services in our models, encompassing the features of a RESTful interface, conditions to invoke methods under right conditions and evaluating the expected results, HTTP requests and response pairs derived from the models, providing hyperlinks that treat hypermedia as engine of application states and provision of a secure web. In the rest of the paper we show how these features are realized in our modeling approach.

To demonstrate our approach, we use as example an imaginary hotel room booking (HRB) service. The service allows a client to book a room, pay for the reservation, and cancel it. It is a simplified pedagogical example, but it shows how to design a REST interface for a service with a complex application state.

The paper is organized as follows. Section 2 gives an overview of the approach and Section 3 discusses the application of be-
havioral interfaces. The conceptual resource model and the behavioral model are presented in Section 4 and 5, respectively. Section 6 shows how these models lead to RESTful interfaces. Section 7 presents the generation of contracts from models and behaviorally enriched WADL. In section 8, we discuss our RESTful behavioral interfaces. Section 10 presents the related work and Section 11 provides conclusion of the paper along with the future work.

2. APPLICATIONS OF BEHAVIORAL INTERFACES

Service descriptions are often used to automatically generate code stubs to invoke the service from a particular programming language. Another interesting application is the creation of a service register to publish and discover web services. Still, a service description enriched with behavioral contracts has many other applications that we describe below.

A web service developer can use the behavioral REST interface as a specification to implement the web service. The UML protocol state machines do not contain executable actions, unlike behavioral state machines, and hence are not executable. On the other hand, they provide rich behavioral information of an interface. Developers implementing a web service have to manually implement the interface specified in protocol state machine. This requires efforts to ensure that implementation conforms to its behavioral specification. Our approach can be extended to generate implementations from the presented models in an automated manner. In addition, behavioral specifications can be automatically generated from models and asserted as contracts into programmatic interface of the web service. This work is under development using Django framework.

A service implementation can use the asserted contracts to validate a request from the client. The preconditions of a method provide a check on the incoming request. Thus, ensuring whether the conditions to invoke a service method are met before invoking the method can be an efficient activity in terms of cost and bandwidth. Similarly, a client can benefit from the asserted postconditions to validate a response from the server, constraining the provider of the service to ensure the functionality that is expected from it.

Our approach can generate implementation stubs and also extract information from behavioral model of a web service and assert it in syntactic interface of a web service. The extended WADL descriptions provide behavioral interface specifications in a machine-processable format.

In order to validate a service, test cases can be generated from the behaviorally enriched interfaces. These test cases can validate the implementation of a web service and these contracts can also be exploited for the generation of test oracles. Test oracles are used to determine whether a test has passed or failed. In the context of test case generation and test oracle generation, we can take advantage of several efforts done previously to validate the behavior of classes and web services using contracts [8] [7].

We can also use these behavioral web service interfaces to provide a monitoring mechanism. The behavioral specifications can be added as a proxy interface to already developed and deployed web services to monitor their functioning. This facilitates location of the fault in and application by observing the conditions that are not being met and by which methods. It can also check for any failure caused by a network fault, late delivery or if a service developer violates a certain pre or post condition of a method by mistake during implementation.

Finally, users of a web service can use the models and contracts as detailed documentation on how to use a service correctly.

In the next sections we present in detail how behavioral interface for RESTful web services are designed in our approach.

3. OVERVIEW OF THE APPROACH

We present an approach that creates RESTful web services by construction. We use UML class diagrams and UML protocol state machines with state invariants to model behavioral REST web service interface.

The structural and behavioral UML models proposed in this article for a web service comply with the requirements highlighted in the previous section and lead to RESTful interfaces with RMM maturity level 3.

UML is a standard modeling notation that provides representation of the system in an abstract manner from different perspectives. It also serves as part of the specification document.

The starting point of our approach is an informal web service specification in natural language. This specification is used to create a conceptual resource model and a behavioral model of the web service. The conceptual resource model is represented diagrammatically using a UML class diagram. It represents the resources involved in a service and their connections and it tackles the addressability and connectedness requirements of the REST style. The behavioral model of the service is represented using a UML protocol state machine with state invariant and extensions to support HTTP request parameters and participants of the service. A protocol state machine contains a number of states and transitions. Each transition is triggered by a method and each state should have an invariant. By constraining the allowed transition triggers to the standard HTTP method we comply with the uniform interface requirement and by defining the state invariants using addressable resources we comply with the statelessness requirement. The outgoing transitions from each state provide information on hyperlinks to further invocations that comply with the hypermedia constraint and labeling of participants as actors facilitate the authentication mechanism.

A contract specifies the pre- and postconditions for the methods of a class. The role of contracts as behavioral interfaces has been investigated for software classes [14, 7, 5] and also in the domain of web services [10, 8]. A contract binds user of the service to pose a valid request and constrains its provider to provide the correct behavior. We show how the pre-and
postcondition of each service request can be generated from the proposed UML models and how these pre- and post conditions can provide behaviorally enriched WADL interface.

The service can be described using the UML language or using the WADL extended with preconditions and postconditions. UML models facilitate the design process as they can be read and understood by human developers with relative ease. The WADL descriptions are in a machine readable format that can be processed by machines for extracting the required information.

4. CONCEPTUAL RESOURCE MODEL

The concept of a resource is central to Resource Oriented Architecture. Any important information in a service interface is exposed as a resource. A resource is something that can be referred to and can have an address. Resources in a resource oriented architecture are analogous to objects in object-oriented paradigm \[17\] or entities in the entity-relationship model.

However, unlike objects and classes in OO paradigm, the resources in resource oriented paradigm are designed with the mind set to expose relevant information for manipulation instead of hiding it \[18\]. Also, in a RESTful interface, resources do not have different access methods, instead the standard HTTP methods are used. Our approach uses four HTTP methods i.e, GET, PUT, POST and DELETE, for retrieving and updating data in a resource.

We use UML class diagram to represent the structure and connections of the resources in a RESTful web service. A UML class diagram represents mainly classes and their associations. An association defines a relationship between two classes by which one class knows about the other class \[19\].

The complexity of a service can be reduced by increasing the number of resources. This results in decoupling of information. A resource can also be a collection resource that contains a group of other resources. In our context this requirement is fulfilled if each resource has a cardinality of more than 1 on the association end of a child resource. A GET method on a collection resource returns a list of all the child resources it contains. For example, a GET method on bookings will give a list of all the booking resources that it contains.

Collection resources are also used as the starting point of the paths to address each resource. Starting from a collection resource, we can access other resources by navigating the successive associations. Paths visiting the same association more than once are not valid. In our example, the following paths are valid.

The REST style requires that all resources should be addressable. In our context this requirement is fulfilled if each resource can be reached from at least one collection resource by navigating one or more associations.

4.2 Collection Resources and Addressability

In Figure 1, bookings and rooms represent collection resources with the stereotype collection and are linked to child resources, booking and room, respectively. A collection resource has a cardinality of more than 1 on the association end of a child resource. A GET method on a collection resource returns a list of all the child resources it contains. For example, the following paths are valid.

The REST style requires that all resources should be addressable. In our context this requirement is fulfilled if each resource can be reached from at least one collection resource by navigating one or more associations.

4.3 Methods

A UML class diagram allows us to define a number of operations for each class. Since a RESTful web service provides uniform interface for all resources, most classes would only have from one to four methods names GET, POST, PUT and DELETE. We do not consider necessary to add this information in the conceptual model.

5. BEHAVIORAL MODEL

The purpose of the behavioral model is to describe the behavior interface specifications of a RESTful web service. It shows the sequence under which operations should be invoked, the parameters passed along with the HTTP requests, the actors that invoked these methods, the conditions under which they can be invoked and the expected results.

Since we are describing RESTful web interfaces, the only allowed operations are GET, POST, PUT and DELETE on resources.

The GET method retrieves a representation of a resource and it should not have side effects. Due to the addressability requirement, it should be always possible to invoke a GET method over a resource. For example, GET/(bookings/{bid}/payment) and GET/(bookings/{bid}/rooms/{rid}) are HTTP
GET methods on the resources payment and room, respectively. Whenever a GET method is called on a resource, it gives the representation of resource as a response. In practice, the access to resources may be restricted by an authentication and access control mechanism.

The POST, PUT and DELETE methods can have side effects. In our hotel booking example, one of the operations of the service is to pay a booking. This is achieved by a HTTP POST request over a payment resource. However, a payment can only be accepted if it is connected to a room booking and a booking can only be paid once. Also, a booking can be canceled, but not while the payment is being processed by a financial service. We need to define all these conditions in the behavioral interface of the service.

We propose to use a UML protocol state machine with state invariants to describe the allowed operations in a web service. We consider that a UML protocol state machine is suitable for representing the behavior of a web service as it provides interface specifications with information on conditions under which methods can be invoked and the expected output from them.

A UML protocol state machine contains mainly states and transitions. We require that each state has a state invariant that is defined as a boolean expression. We then say that a state is active if and only if its state invariant evaluates to true. A state may contain other states and is called a composite state. In such a case, the actual state invariant of the contained state is given by the conjunction of the state invariant specific for the contained state and the state invariants of all the states that contain it. These state invariants within a composite state should be mutually exclusive. That is, only one state within a region of a composite state can be active at a time.

A transition is an arc from one or more source state(s) to one or more target state(s) labeled with a method name and a guard. If the source states are active, the guard is true and the method is invoked, then the transition may be fired and as a consequence the target state(s) become active. When no guard is shown in the transition it is assumed to be true.

The only allowed method in our state machine are GET, PUT, POST and DELETE to provide a uniform interface. These methods are represented as: \textit{METHOD\_NAME (path, p1, p2, ...)}, where \textit{path} is the relative path of the resource on which the HTTP method is invoked, and \textit{p1, p2, ...} represent the HTTP request parameters, if any, passed with the method.

In our behavioral model, the transition triggers can only be defined as POST, PUT or DELETE operations over resources described in the conceptual model. The guards and the state invariants can be defined only using information from the resources and request parameters.

When we invoke an HTTP GET method on a resource, it returns its representation along with the HTTP response code. This response code tells whether the request went well or bad. If the HTTP response code is 200, this means that the request was successful and the referred resource exists. Otherwise, if the response code is 404, this implies that URI could not be mapped to any resource and the referred resource does not exist.

We manipulate this information exhibited by the response codes and representation of resources by comparing them with the expected behavior. We use a boolean function \textit{OK(r)} to express that the response code of HTTP GET method on resource \textit{r} is 200. Similarly, the boolean function \textit{NOT\_FOUND(r)} is true when the response code of HTTP GET method on resource \textit{r} is 404. These boolean functions on the resources along with the attributes that represent a resource are used to define a state invariant in our RESTful behavioral model.

For example, consider the state invariant for the state \textit{reserved\_not\_paid} in Figure 2. \textit{NOT\_FOUND(p)} checks the response code for the HTTP GET method on the resource payment. It evaluates to true if response code of GET method
on \( p \) for a particular booking ID(\{bid\}) is 404. For the HRB service to be in state \( \text{reserved\_not\_paid} \), the state invariant of this simple state is conjuncted with the state invariants of all the states that contain it.

Our behavioral model shows different states of a RESTful web service and gives information on what HTTP methods on a particular resource can be invoked from a certain state. According to Figure 2, the protocol state machine of HRB service is initiated by the HTTP POST method on the bookings resource. This creates a booking resource with a specific booking ID in the collection resource bookings. The initial state Booking is a composite state with two more composite states (\( \text{reserve\_and\_pay} \), \( \text{pconfirmation\_info} \)) and one simple state (\( \text{waiting\_for\_payment} \)). When the room is booked and the payment is made, the service waits for the confirmation of the payment. The state \( \text{waiting\_for\_payment} \) has a self transition on trigger \( \text{POST}\(/\text{bookings/bid}/\text{payment/}\text{pwating} \) under the conditions that the payment is not confirmed and the number of tries to confirm the payment is less than or equal to three. If the payment fails to confirm even after three tries, then the information about an unconfirmed booking is given. Otherwise, if the payment is confirmed then it gives the booking confirmation.

Whenever a PUT, POST or DELETE method is called on a resource it changes the state of the application in the behavioral model. A booking can be canceled from composite states \( \text{reserve\_and\_pay} \) and \( \text{confirmation\_info} \) and a booking can be deleted only if it is canceled. A booking cannot be canceled if it is waiting for the payment confirmation from a third-party service. The request parameters passed along with the HTTP request are those attributes of the resource on which user’s input/data is required. For example, a POST on payment resource requires request parameters \( \text{ccName} \) and \( \text{ccNumber} \) to make a method invocation, i.e., \( \text{POST}\(/\text{bookings/bid}/\text{payment, ccName, ccNumber} \).

A guard condition on the transition specifies the condition required to invoke an HTTP method on a resource. For example, consider guard \( \{\text{pc.confirm==true}\} \) for the method \( \text{POST}\(/\text{pconfirmation} \) in Figure 2, where \( \text{pc} \) refers to the relative navigation path to resource \( \text{pconfirmation} \). This guard specifies that the POST method on \( \text{pconfirmation} \) can be invoked if the resource representation of \( \text{pconfirmation} \) contains \( \text{true} \) value for the \( \text{confirm} \) attribute.

5.1 Stateless State Machines?

Using a state machines to model a stateless interface may seem an oxymoron. In the context of a RESTful service, statelessness is interpreted as the absence of hidden information kept by the service between different service requests. In that sense, a RESTful web service should exhibit a stateless protocol. Also, there is no sense of session or sequence of request in a true RESTful service.

On the other hand, state machines have a notion of active state configuration, that is, what states are active at a certain point of time. If an implementation of an interface described using a state machine would have to keep the active state configuration between different requests, then this would break the statelessness requirement of the RESTful service.

However, our approach does not actually require that a service implementation keeps any additional protocol state. In our approach a state is active if its invariant evaluates to true, but the invariants are defined using addressable appli-
6. FROM SERVICE MODELS TO A REST INTERFACE

Our behavioral model of RESTful web service interface is rich in behavioral information and every transition provides manifold information. This information contributes to the creation of web services that exhibit pure RESTful behavior with provision to validate whether the implementation conforms to its design specifications.

5.2 Authorization and Actors

In a secure web, the requests must be authenticated to ensure that the request is coming from the right party and if the consumer of the service has the right to access the privileged resource. In this context, our behavioral model introduces the notion of actors in our behavioral model that specify which party invokes a certain method.

An actor of the service is the participant that invokes different methods on different resources of the service resulting in different application states. In Figure 2, the involved parties are annotated as actors on the transitions along with the methods they trigger, guards and request parameters. The `client` invokes POST on `bookings` and `payment` resources, where as the `HRBService` invokes the third party payment service `waiting` resource for an asynchronous third party payment service. A POST on `confirmation` resource is invoked by the `PaymentService`. A booking can be canceled or deleted by the `client`.

When a consumer of the service makes an HTTP request on privileged resource, it provides its credentials in the Authorization header. If the credentials are wrong or not provided, consumer is denied access to the resource. Thus, the specification of involved parties in the behavioral model along with the methods they invoke facilitate the developer in creating secure RESTful web service and the developer can use this information when implementing the service.

A transition can occur from one state to another if the method that triggers this transition is invoked and its precondition is true. For the transition to be successful, the postcondition of the transition should also be true after the transition.

The precondition of a method states under what conditions a method can be triggered. We say that the precondition of a method \( m \) is satisfied when the state invariants of all the source states of transition \( t \) are true along with its guard condition.

In a similar manner, if a method \( m \) triggers a transition \( t \) in a behavioral model, then its post-condition is satisfied when the state invariants of all the target states of transition \( t \) are true along with the postcondition annotated on the transition \( t \).

We use the formalization of the structure and semantics of the UML protocol state machine presented in [15]. This formalization supports formal definitions for generating pre- and post-conditions for class methods. We extend this work by generating contracts for HTTP methods from our behavioral models for RESTful web services.

In order to shorten the description of the contract we use path variables to represent the address of a resource. First, the precondition for a method that triggers a transition in the behavioral model is presented. The precondition of a method \( m \) is given by taking into account all the transitions that are triggered by \( m \). If it is a simple transition, then the state invariant of its source state is conjuncted with the guard of the transition. In case the transition is a trigger to more than one transition, with true guards, and all the transitions have different source states, then the precondition is given by taking a disjunction of state invariants of all the different source states. This implies that the method can trigger a transition whenever it is in one of its source states.

A transition can occur from one state to another if the method that triggers this transition is invoked and its precondition is true. For the transition to be successful, the postcondition of the transition should also be true after the method is invoked. This is specified by the implication operator that relates a precondition of a transition with its postcondition.

A postcondition for a method is extracted from the protocol state machine by manipulating the state invariants of the target states of transitions and the post-conditions on transitions. The post-condition of a fork transition, with true postcondition, specifies that the state invariants of all its target states are true and for a self-transition, its postcondition ensures that the same state invariants are true that were true before invoking the HTTP method.

For the details and formal definitions of generating preconditions and postconditions for different elements in a UML protocol state machine of a class readers are referred to [15].

The postcondition of a transition will be evaluated only if the precondition for that transition is true. We define as \( \text{pre}_\text{OK}(r) \) the function that gives boolean value of \( \text{OK}(r) \) on resource \( r \) before invoking the trigger method. Similarly, \( \text{pre}_\text{pc.waiting}, \text{pre}_\text{pc.confirm} \) and \( \text{pre}_\text{NOT\_FOUND}(r) \) give the representation of `confirmation` and boolean value of \( \text{NOT\_FOUND}(r) \) before invoking the trigger method, respectively.

The precondition of a method states under what conditions
The excerpt below from the list of high-level contracts generated from Figure 2 shows the contracts generated for the HTTP method POST on cancel resource.

```
<table>
<thead>
<tr>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>b: bookings/{bid}/</td>
</tr>
<tr>
<td>r: bookings/{bid}/rooms/{rid}</td>
</tr>
<tr>
<td>pc: bookings/{bid}/payment/pconfirmation</td>
</tr>
<tr>
<td>e: bookings/{bid}/cancel</td>
</tr>
</tbody>
</table>

| POST | bookings/{bid}/cancel |
|------|
| precondition |
| (OK(b) & & NOT_FOUND(c)) & & (NOT_FOUND(pc) & & NOT_FOUND(pw)) | | (OK(pc) & & NOT_FOUND(pc))) |
| postcondition |
| (((pre_OK(b) & & pre_OK(r) & & pre_NOT_FOUND(c)) & & (pre_FOUND(pc) & & pre_FOUND(pw)) | | OK (b) & & OK(c)) & & (pre_OK(pc) & & pre_FOUND(pw)) | | OK (b) & & OK(c)) |

The conceptual model as shown in Figure 1 and behavioral model as shown in Figure 2 lead to rich RESTful interface descriptions as explained in Section 7.

6.2 HTTP Requests and Responses

The behavioral model in Figure 2 shows the HTTP methods that can be invoked on different resources and the order in which they should be invoked to make a hotel booking. When an HTTP request (allowed method) is made on a resource, its HTTP response is sent with entity headers along with the representation of the resource in that state (if content exists). The conceptual model in Figure 1 shows the attributes of each resource that become part of its representation. In addition, the representation will contain links mapped to the outgoing transitions from the state. These links or outgoing transitions provide information on what further links can be taken by the client to fulfill the goal of making a booking. We can use this information to create links in our representation creating RESTful web service that comply with Level 3 of Richardson Maturity Model.

Table 1 shows the request and response pairs for the conceptual and behavioral models of HRB RESTful web service. The HTTP requests on the resources contain request parameters that map to the parameters on transitions in Figure 2. The HTTP responses contain the HTTP response code and representation of the resource. The representation of the resource contains attributes that map to the resource representations in the conceptual model in Figure 1 and the links contain the relative paths of the outgoing transitions from the target state of the transition as in Figure 2. We represent link element in JSON by two attributes **href** and **rel** as defined in [1]. The **rel** attribute stands for relations and indicates the type of link. The **href** attribute contains the link’s URI and **rel** contains the name of the resource to which the hyperlink points to. It is a good practice to provide HTML documentation at the URI that contains information on purpose of link relation, valid HTTP methods for the target URI and expected media types[1].

As an example, lets take a POST request on payment resource in Table 1. The POST request is invoked on payment resource with request parameters **ccName** and **ccNumber**. Its HTTP response contains response code 200, JSON representation of updated payment resource and links to **pwaiting** and **cancel** resource referring to transitions that invoke payment service for payment processing and for canceling the booking. The array of links also contains other allowed links that can be traversed from this application state, e.g., the booking resource can be traversed from here, but it does not contain any link for payment confirmation resource that cannot be invoked until the payment has been processed. The developers can implement the authorization headers using the actor information on transitions to ensure that right party is making the right request on a privileged resource. Similarly, when a payment is waiting for payment confirmation and POST on **pwaiting** is invoked, its representation in HTTP response does not contain link to cancel the booking as a booking cannot be canceled if it is waiting for payment confirmation. A DELETE on booking returns an HTTP response code of 204 implying there is no content available and hence no further links. Thus, with our behavioral model the sequence of method invocation is preserved along with their application states and also the stateless feature is not compromised. In addition, in compliance with RMM, our behavioral model creates web services with hypermedia as an engine of application states.

7. GENERATION OF BEHAVIORAL WADL SERVICE DESCRIPTIONS

Web Application Description Language (WADL) provides a description language for RESTful web services. It is used for publishing RESTful web service interfaces and provides machine-processable description of the interface without cluttering it with too many details [9]. Currently, RESTful architectural style and WADL are being widely adopted in the web and have numerous users, including enterprises such as Google, Yahoo, Amazon and Flickr.

The information about the interface of a web service in WADL is syntactic and does not state anything about its semantics, how a service should be invoked and behave. Also, WADL is not actually restricted to RESTful services and can be used to describe services that do not follow this architectural style completely.

Besides the UML representation of the behavior interfaces, we propose to extend the Web Application Description Language (WADL) to include information about the behavior of the methods in a service. Our objective is that this information is generated automatically from the conceptual and behavioral models described before.

WADL defines the operations that can be invoked on an interface and describes the input and output parameters for each operation. It defines the resources that an application contains and methods that can be called on them. Each method has two attributes **name** and **id**, where **name** is the name of the HTTP method and **id** is the ID of the method that is associated with the HTTP method.

Representing the information in the conceptual model as part of a WADL service description is a rather straightforward task. However, the behavioral model does not map
<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST /bookings/ HTTP/1.1</td>
<td>HTTP/1.1 201 Created</td>
</tr>
<tr>
<td>Host: <a href="http://www.example.org">www.example.org</a></td>
<td>Location: <a href="http://www.example.org/bookings/021">http://www.example.org/bookings/021</a></td>
</tr>
<tr>
<td>Content-Type: application/json</td>
<td>Content-Type: application/json</td>
</tr>
<tr>
<td>{‘guestID’: ’42’,’roomID’: ’34’ }</td>
<td>{‘bid’: ’021’, ’bdate’: ’20-11-2009’, ’guestID’: ’42’,’roomID’: ’34’ }</td>
</tr>
<tr>
<td></td>
<td>’link’:[{’rel’: ’payment’, ’href’: ’/bookings/021/payment/’}, {’rel’: ’cancel’, ’href’: ’/bookings/021/cancel/’}]</td>
</tr>
</tbody>
</table>

| POST /bookings/021/payment/ HTTP/1.1 | HTTP/1.1 200 OK |
| Host: www.example.org | Location: http://www.example.org/bookings/021/payment/ |
| Content-Type: application/json | Content-Type: application/json |
| {’ccName’: ’Richard’,’ccNumber’: ’142-567-719’} | {’pid’: ’10’, ’bookingid’: ’021’, ’amount’: ’350’ } |

| POST /bookings/021/payment/pwaiting/ | HTTP/1.1 200 OK |
| Host: www.example.org | Location: http://www.example.org/bookings/021/payment/pwaiting/ |
| Content-Type: application/json | Content-Type: application/json |

| POST /bookings/021/payment/pconfirmation/ | HTTP/1.1 200 OK |
| Host: www.example.org | Location: http://www.example.org/bookings/021/payment/pconfirmation/ |
| Content-Type: application/json | Content-Type: application/json |
| | {’confirm’: True’, ”link’:[{’rel’: ’cancel’, ’href’: ’/bookings/021/cancel/’}, {’rel’: ’booking’ ’href’: ’/bookings/021/’}] } |

| POST /bookings/021/cancel/ HTTP/1.1 | HTTP/1.1 200 OK |
| Host: www.example.org | Location: http://www.example.org/bookings/021/cancel/ |
| Content-Type: application/json | Content-Type: application/json |
| | {’cancel’: True’, ”link’:[{’rel’: ’cancel’, ’href’: ’/bookings/021/cancel/’}, {’rel’: ’booking’ ’href’: ’/bookings/021/’}] } |

| DELETE /bookings/021/ HTTP/1.1 | HTTP/1.1 204 No Content |
| Host: www.example.org | |
directly to a WADL description since the behavioral model allows different transitions to be trigger by the same method. In our example, a cancel request can be invoked when the application is in different states. That is the information about when a method can be invoked (precondition) and what is its result (postcondition) needs to be computed from the different states in the behavioral model.

7.1 Inserting Pre- and Postconditions into WADL Service Descriptions

The high-level contracts are refined with details on relative navigation paths, invoked HTTP methods and the expected response codes. These refined contracts are asserted into WADL interface. The function \( \text{pre}_\text{OK} (r) \) is mapped to a \( \text{GET} \) function and its response code is compared to 200. The \( \text{pre}_\text{GET} (r) \) function gives the stored results of invoking a GET method on resource \( r \) before invoking the method. In similar manner, a \( \text{pre}_\text{NOT}_\text{FOUND} \) is mapped to a \( \text{GET} \) function and its response code is compared to 404.

In order to support the behavioral information in the interface descriptions, we have extended the XML schema of WADL with two elements \( \text{precondition} \) and \( \text{postcondition} \), with an attribute \( \text{id} \) for each of these elements. These tags, i.e., \( \langle \text{precondition} \rangle \) and \( \langle \text{postcondition} \rangle \) are asserted above and under the method tag, respectively. An excerpt of a behavioral RESTful interface is shown below for method POST on \( \text{cancel} \) resource.

```xml
<resources base = "http://www.example.com/bookings">
  ...
  <resource path = "{bid}">
    ...
    <precondition id = "pre_post_cancel">
      <GET>/bookings/{bid}/rooms/{rid} = status(200) &
      <GET>/bookings/{bid}/payment/pconfirmation = status(404) ||
      <GET>/bookings/{bid}/payment = status(200) &
      <GET>/bookings/{bid}/cancel = status(404)
    </precondition>
    <method name = "POST" id = "cancel">
      ...
      <postcondition id = "post_post_cancel">
        (pre_GET'/boookings/{bid}/) = status(200) &
        <GET>/bookings/{bid} = status(200) &
        <GET>/bookings/{bid}/cancel = status(404) &
        (pre_GET'/bookings/{bid}/payment/pconfirmation = status(404)
          || GET'/boookings/{bid} = status(200) &
          (GET'/bookings/{bid}/cancel = status(404))
      </precondition>
    </method>
  </resource>
</resources>
```

8. RESTFUL BEHAVIORAL INTERFACES

As discussed earlier, a REST interface should exhibit these four attributes: \textit{addressability}, \textit{connectivity}, \textit{statelessness} and \textit{uniform interface}. We claim that we can create RESTful interface following the approach described in this paper. Our conceptual resource model and behavioral model lead to web services that exhibit these attributes and make our interfaces RESTful by construction.

We constrain our conceptual model to be a connected graph such that no resource is isolated. The associations between classes in the conceptual model provide \textit{connectivity} between the resources. Each resource can be addressed independently using the navigation directions of associations and their role names. The role names show the relative navigation path between the resources. For example, the navigation path to a room belongs to a particular booking ID(bid) is given as \( /\text{bookings/\{bid\}/rooms/\{rid\} } \). Thus, each resource has a URI address providing \textit{addressability} feature to our conceptual resource model.

Unlike software class diagrams showing static structure of RPC web services, our conceptual model does not contain any method information. We restrict the behavioral model so that transitions can only be triggered using the standard HTTP methods, providing the \textit{uniform interface} feature.

In addition, we have created stateful service using stateless service interface thanks to the fact that the application states are defined using state invariants that are defined in terms of exposed resources. This information is presented in the behavioral model of the REST interface, providing the \textit{statelessness} feature in our behavioral model.

Another strength of our design approach is that the web services are created RESTful by construction with maturity of level 3 services. Our behavioral model is rich in behavioral specifications, providing addressable resources, allowed HTTP verbs on them and the conditions under which they can be invoked. In addition to that, conforming to level 3 of RMM, behavioral model contains outgoing transitions from states that specify what further transitions can be taken from these states. This information can be represented as links in the response of HTTP request. This leads consumers through trail of resources resulting in application state transitions.

We also enrich the RESTful behavioral interface with information that can provide clear mapping to HTTP requests and responses. Transitions are labeled with request parameters that are passed along with the HTTP request and the party that invokes the method. This information can then be used to authenticate requests by the consumers on the privileged resources and generated appropriate HTTP requests.

9. RELATED WORK
Several authors have used UML class diagram and other UML behavioral diagrams to model RPC styled web services and their compositions [16].

In terms of modeling behavioral specifications of web services, Bertolino and Polini have used UML protocol state machines for specifying the intended protocol of services [2]. Protocol state machines are transformed to Symbolic Transition System that is extended with Labeled Transition System(LTS) semantics.

Modeling of RESTful web services has been addressed in the work of Sista and her research group [11, 12, 18]. In [11, 12], an approach that migrates legacy APIs to RESTful web services is presented. In [12], Laitkorpi et al present the process that transforms service functionality over different phases to a RESTful service. This approach is further explored in [18] by Siikarla et al. They provide an iterative and incremental process for the development of model transformations by focusing on the step of transforming information model to resource model.

The role of contracts in the domain of web services has been investigated previously, e.g., [6] [4], etc. In [6], Castagna et al present theory of contracts that formalizes the compatibility of a client to a service and introduces a subcontract relation for behavioral typing of web services promoting service reuse or redefinition. In [3] and [4], a theory of contract is presented that addresses the problem of composition of multiple services. The correctness for service compositions is modeled using process calculi and the notions of strong service compliance and strong subcontract pre-order are investigated.

For REST architectural style, Zou et al [21] propose an Accountable State Transfer(AST) architecture to bridge the accountability gap in REST. The approach uses service contract as foundation for enabling accountability. The paper outlines the architectural decisions and principles for enabling accountability, followed by AST architecture with accountable state transfer protocol. The service contracts are implemented as an ontology and the service executions are stored in a knowledge base to allow reasoning between different concepts of a contract.

In the context of modeling behavioral specifications and using contracts with UML, Lohmann et al [13, 10], use visual contracts to specify the dynamic behavior and class diagrams to specify the static aspect of a web service. Graph transformations are annotated on to the class diagram with object diagrams specifying pre- and post-conditions of the operations.

We consider that the main strength of our approach is that our models lead to RESTful web services. In addition, we can generate implementation stubs and behaviorally enriched interfaces from models that have many applications. The use of UML protocol state machine for modeling provides behaviorally enriched interfaces that show the sequence of operations that can be invoked on the service, the conditions under which they can be invoked and the expected results. Both the conceptual and behavioral models serve as specification document. Also, contracts can be directly generated from these models and asserted into syntactic service descriptions as shown in Section 4 and 5.

10. CONCLUSION AND FUTURE WORK

In this article, we present a novel approach to model the behavioral interface of a RESTful web service. A conceptual resource model shows different resources and how can they be addressed. The behavioral model specifies how the service should be used by showing the order of method invocations and the conditions on these methods. In addition, it provides outgoing links after a particular method invocations leading to a trail of resources that provide application states. It is also enriched with information on participants of the web service and request parameters providing authentication mechanism and mapping to HTTP requests and responses.

We have used UML class and protocol state machine diagrams to model the conceptual and behavioral aspects of the web service. The models can be used to generate a contract in the form of preconditions and postconditions for methods of an interface. These contracts can be included in a WADL interface specification. The resulting service is RESTful by construction.

The behavioral RESTful interfaces have many applications. They can serve as documentation for existing services or as a blue print for developing new ones. The models can be used to generate implementation stubs in commonly used web frameworks like Django and Ruby on Rails. They can also be used to monitor the interaction between a service and its clients and warn if any of the parties breaches the interface contract. The generation of tests from interface contracts is also a promising application of formalized behavioral interfaces.

In the context of web services, our work is novel as we provide a modeling approach for a RESTful web service that captures both the static and behavioral features of an interface. The models exhibit all attributes of REST web services with maturity of level 3 of Richardson Maturity Model. They are relatively easy to understand and communicate.

We are currently working on adding semantic capabilities to our models and providing a mechanism to validate semantic web services. We are also interested in extending our approach to support RESTful web service composition.

11. REFERENCES


About the authors:

Irum Rauf is a PhD candidate focusing on design, validation and verification of RESTful web service. She works at the Department of Information Technology at Abo Akademi University in Finland where she is involved in different research and teaching activities. Previously, she has done her Masters in Software Engineering from Muhammad Ali Jinnah University Pakistan and has been involved in different teaching tasks. She holds a funded position in Software Systems and Engineering (SoSE) Graduate school. Earlier, she has also worked as Manager Web Services in Teralight(Pvt.) Ltd, Islamabad and Project Manager and Management Representative (ISO9000) for Pakistan Software Solutions (Pvt.) Ltd., Pakistan. She has also been awarded Nokia's scholarship award for the year 2011.

Ivan Porres is a Professor in Software Engineering and head of the Computer Engineering education at the Department of Information Technologies at Åbo Akademi University, in Turku, Finland. He is also the leader of the Software Engineering Laboratory at TUCS, the Turku Center for Computer Science.

His research interests include model-based software and system engineering, including modeling languages and design environments, and their applications. He is also interested in the research of software processes and software process improvement approaches. He has published more than 75 articles on these topics. He serves as a reviewer and member of program committee in numerous journals and conferences related to software and system modeling.
ABSTRACT

Data Web services allow users to access information provided by different companies. Web users often need to compose different Web services to achieve a more complex task that cannot be fulfilled by an individual Web service. In addition, user preferences are becoming increasingly important to personalize the composition process. In this paper, we propose an approach to compose data Web services in the context of preference queries where preferences are modeled thanks to fuzzy sets that allow for a large variety of flexible terms such as “cheap”, “affordable” and “fairly expensive”. Our main objective is to find the top-k data Web service compositions that better satisfy the user preferences. The proposed approach is based on an RDF query rewriting algorithm to find the relevant data Web services that can contribute to the resolution of a given preference query. The constraints of the relevant data Web services are matched to the preferences involved in the query using a set of matching methods. A ranking criterion based on a fuzzyfication of Pareto dominance is defined in order to better rank the different data Web services/compositions. To select the top-k data Web services/compositions we develop a suitable algorithm that allows eliminating less relevant data Web services before the composition process. Finally, we evaluate our approach through a set of experiments.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Query formulation, Search process, Selection process; H.3.5 [Online Information Services]: Web-based services

General Terms

Measurement, Theory, Performance

Keywords

Fuzzy dominance, Web service, Top-k compositions, fuzzy preferences queries.

1. INTRODUCTION

Nowadays, an increasing number among companies are moving towards a service-oriented architecture for data sharing on the Web by putting their data sources behind Web services, thereby providing a well-documented and interoperable method of interacting with their data [8, 16, 9]. In particular, if no single Web service can satisfy the functionality required by the user, there should be a possibility to combine existing services together in order to fulfill the user request. In this context, we talk about Data Web Service Composition, where services correspond to calls over the data sources, i.e., the invocation of a data Web service results in the execution of a query over the data sources’ schema.

On the other hand, user preferences are becoming increasingly important to personalize the composition process. For example, when looking for items to be purchased over the Web, customer preferences are critical in the search. A more general and suitable approach to model preferences is that based on fuzzy sets theory [13]. Fuzzy sets are very well suited to the interpretation of linguistic terms, which constitute a convenient way for a user to express her/his preferences. For example, when expressing preferences about the price of a car, users often employ fuzzy terms like “rather cheap”, “affordable” and “not expensive”. However as data Web services and service providers proliferate, a large number of candidate compositions that would use different -most likely competing- services may be used to answer the same query. Hence, it is important to set up an effective service composition framework that would identify and retrieve the most relevant services and return the top-k compositions according to the user preferences.

The following example illustrates a typical scenario related to our previous discussion, showing the challenges in finding the top-k service compositions.

Example. Let us consider an e-commerce system supporting users to buy cars. The system has access to the set of services described in Table 1. The symbols “$” and “?” denote inputs and outputs of data services, respectively. Services providing the same functionality belong to the same service class. For instance, the services $S_{21}$, $S_{22}$, $S_{23}$ and $S_{24}$ belong to the same class $S_2$. Each service has its (fuzzy) constraints on the data it manipulates. For instance, the cars returned by $S_{21}$ are of cheap price and short warranty.

Assume that a user Bob wants to buy a car. He sets his preferences and submits the following query $Q_1$: “return the French cars, preferably at an affordable price with a warranty around 18 months and having a normal power with
Table 1: Example of Data Web Services

<table>
<thead>
<tr>
<th>Service</th>
<th>Functionality</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{11}(x, y)$</td>
<td>Returns the automakers $y$ made in $x$</td>
<td>$-$</td>
</tr>
<tr>
<td>$S_{21}(x, y, z, t)$</td>
<td>Returns the cars $y$ along their prices $z$ and warranties $t$</td>
<td>$z$ is cheap, $t$ is short</td>
</tr>
<tr>
<td>$S_{22}(x, y, z, t)$</td>
<td>for a given automaker $x$</td>
<td>$z$ is accessible, $t$ is long</td>
</tr>
<tr>
<td>$S_{23}(x, y, z, t)$</td>
<td>$z$ is expensive, $t$ is long</td>
<td></td>
</tr>
<tr>
<td>$S_{24}(x, y, z, t)$</td>
<td>$[9000, 14000]$, $t = [0, 24]$</td>
<td></td>
</tr>
<tr>
<td>$S_{31}(x, y, z)$</td>
<td>Returns the power $y$ and the consumption $z$</td>
<td>$y$ is weak, $z$ is small</td>
</tr>
<tr>
<td>$S_{32}(x, y, z)$</td>
<td>for a given car $x$</td>
<td>$y$ is ordinary, $z$ is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>approximately 4</td>
</tr>
<tr>
<td>$S_{33}(x, y, z)$</td>
<td>$y$ is powerful, $z$ is high</td>
<td></td>
</tr>
<tr>
<td>$S_{34}(x, y, z)$</td>
<td>$[60, 110]$, $z = [3.5, 5.5]$</td>
<td>$-$</td>
</tr>
</tbody>
</table>

a medium consumption”. Bob will have to invoke $S_{11}$ to retrieve the French automakers, he can then invoke one or more of the services $S_{21}, S_{22}, S_{23}, S_{24}$ to retrieve the French cars along with their prices and warranties, finally, he will invoke one or more of the services $S_{31}, S_{32}, S_{33}, S_{34}$ to retrieve the power and the consumption of retrieved cars. This manual process is painful and tedious. It raises the following challenges: (i) how to understand the semantics of the published data Web services to select the relevant ones that can contribute to answering the query at hand; (ii) how to retain the most relevant services (several similar data Web services offer the same functionality but are associated with different constraints) that better satisfy the user’s preferences and (iii) how to generate the best $k$ data Web service compositions that satisfy the whole user query.

**Contributions.** We already tackled the first challenge by proposing a semantic annotation of data Web services and an efficient RDF-based query rewriting approach that generates automatically the data Web service compositions that cover a user query (which does not include any preference constraints) [4]. In this paper, we focus on the second and third challenges. We select the services that cover a part of the query even if their constraints match only partially the user preference constraints. Different matching methods are investigated to compute the matching degrees between the services’ fuzzy constraints and the fuzzy preferences involved in the query. In order to select the most relevant services, a ranking criterion based on a multicriteria fuzzy dominance relationship is proposed. The selected services are then used to find the top-$k$ service compositions that answer the user query.

**Outline.** The rest of this paper is organized as follows. In Section 2, we give a necessary background of our paper, i.e., a reminder on fuzzy sets, that constitutes a basis to our work. In Section 3, we formally describe the studied problem. Section 4 describes the ranking approach of data Web services which is mainly based on the fuzzy dominance relationship. Section 5 is devoted to the top-$k$ data Web service compositions generation method. Section 6 presents the global architecture of our Web service composition-based preference query answering and shows some experimental results. We review related work in Section 7. Finally, Section 8 concludes the paper and outlines some perspectives.

## 2. BACKGROUND ON FUZZY SETS

### 2.1 Basic Notions

Fuzzy set theory was introduced by Zadeh [31] for modeling classes or sets whose boundaries are not quite defined. For such objects, the transition between full membership and full mismatch is gradual rather than crisp. Typical examples of such fuzzy classes are those described using adjectives of the natural language, such as “cheap”, “affordable” and “expensive”. Formally, a fuzzy set $F$ on a referential $X$ is characterized by a membership function $\mu_F : X \rightarrow [0, 1]$ where $\mu_F(x)$ denotes the grade of membership of $x$ in $F$. In particular, $\mu_F(x) = 1$ reflects full membership of $x$ in $F$, while $\mu_F(x) = 0$ expresses absolute non-membership. When $0 < \mu_F(x) < 1$, one speaks of partial membership. $F$ is normalized if $\exists x \in X, \mu_F(x) = 1$. Two crisp sets are of particular interest when defining a fuzzy set $F$:

- the core $C(F) = \{ x \in X \mid \mu_F(x) = 1 \}$, which gathers the prototypes of $F$,
- the support $S(F) = \{ x \in X \mid \mu_F(x) > 0 \}$, which contains the elements that belong to some extent to $F$.

In practice, the membership function associated with $F$ is often of a trapezoidal shape. Then, $F$ is expressed by the quadruplet $(A, B, a, b)$ where $C(F) = [A, B]$ and $S(F) = [A - a, A + b]$, see Figure 1. A regular interval $[A, B]$ can be seen as a fuzzy set represented by the quadruplet $(A, B, 0, 0)$.

**Figure 1: Trapezoidal membership function**

Let $F$ and $G$ be two fuzzy sets on the universe (i.e., referential) $X$, we say that $F \subseteq G$ iff $\forall x \in X, \mu_F(x) \leq \mu_G(x)$. The complement of $F$, denoted by $F^c$, is defined by $\mu_{F^c}(x) = 1 - \mu_F(x)$. The cardinality of $F$ is defined as $|F| = \sum_{x \in X} \mu_F(x)$. Furthermore, $F \cap G$ (resp. $F \cup G$) is defined the following way:

- $\mu_{F \cap G} = \cap(\mu_F(x), \mu_G(x))$ where $\cap$ is a t-norm operator that generalizes the conjunction operation (e.g., $\cap(x, y) = \min(x, y)$ and $\cap(x, y) = x \cdot y$).
- $\mu_{F \cup G} = \cup(\mu_F(x), \mu_G(x))$ where $\cup$ is a co-norm operator that generalizes the disjunction operation (e.g., $\cup(x, y) = \max(x, y)$ and $\cup(x, y) = x + y - x \cdot y$).

As usual, the logical counterparts of the theoretical set operators $\cap$, $\cup$ and complementation operator correspond respectively to the conjunction $\land$, disjunction $\lor$ and negation $\neg$. See [13] for more details.

A fuzzy implication is any $[0, 1]^2 \rightarrow [0, 1]$ mapping $I$ satisfying the boundary conditions $I(0, 0) = 1$ and $I(1, x) = x$.
for all $x$ in $[0, 1]$. Moreover, we require $I$ to be decreasing in its first, and increasing in its second component. Two families of fuzzy implications are studied in the fuzzy community (due to their semantic properties and to the fact that their results are similar with the ones of usual implications, material implications, when the arguments are 0 or 1):

- **R-implications**: they are defined by $I(x, y) = \text{sup}\{\beta \in [0, 1], \top(x, \beta) \leq y\}$, where $\top$ is a t-norm operator. The two most used R-implications are Gödel implication ($I_{Go}(x, y) = 1$ if $x \leq y$, 0 otherwise) and Goguen implication ($I_{Go}(x, y) = 1$ if $x \leq y$, $y/x$ otherwise).

- **S-implications**: they are defined by $I(x, y) = \bot(1 - x, y)$, where $\bot$ is a co-norm operator. The two most popular S-implications are Kleene-Dienes implication ($I_{Kd}(x, y) = \text{max}\{(1 - x, y)\}$) and Lukasiewicz implication ($I_{Lu}(x, y) = \text{min}\{1 - (1 - x + y), 1\}$).

Note that Lukasiewicz implication is also an R-implication. For more details about fuzzy implications, see [13].

### 2.2 Modeling Preferences

Fuzzy sets provide a suitable tool to express user preferences [12][14]. Fuzzy set-based approach to preferences queries is founded on the use of fuzzy set membership functions that describe the preference profiles of the user on each attribute domain involved in the query.

The user does not specify crisp (Boolean) criteria, but gradual ones like “very cheap”, “affordable” and “fairly expensive”, whose satisfaction may be regarded as a matter of degree. Individual satisfaction degrees associated with elementary conditions are combined using a panoply of fuzzy connectives, which may go beyond conjunctive and disjunctive aggregations. Then, the result of a query is no longer a flat set of elements but is a set of discriminated elements according to their global satisfaction w.r.t. the fuzzy conditions appearing in the query. So, a complete pre-order is obtained.

### 3. SERVICE COMPOSITION-BASED PREFERENCE QUERIES ANSWERING

#### 3.1 Preference Queries

Users express their preference queries over domain ontologies using a slightly modified version of SPARQL query language. Preferences are modeled by means fuzzy sets. For instance, query $Q_1$ given in Section 1 is expressed as follows:

**SELECT ?n ?p ?w ?co**

**PREFERING (?pr is 'affordable' ?w is 'around 18', ?pw is 'normal' ?co is 'medium')**

The semantics of all fuzzy terms of Table 1 are available in the URL: [http://vm.liris.cnrs.fr:36880/FuzzyTerms/](http://vm.liris.cnrs.fr:36880/FuzzyTerms/).

#### 3.2 Data Services

Assume that data services are partitioned into different service classes. A class $S_j$ represents services with the same inputs, outputs, and providing the same functionality but with different (fuzzy) constraints. A data service $S_{ji}$ of class $S_j$ is described as a predicate $S_{ji}(\exists X_j, ?Y_j):<\phi_j(X_j, Y_j, Z_j), C_{ji}>$ where:

- $X_j$ and $Y_j$ are the sets of input and output variables of $S_{ji}$, respectively. Input and output variables are also called distinguished variables. They are prefixed with the symbols “$” and “?”, respectively.
- $\phi_j(X_j, Y_j, Z_j)$ represents the functionality of the service. This functionality is described as a semantic relationship between input and output variables. $Z_j$ is the set of existential variables relating $X_j$ and $Y_j$.
- $C_{ji} = \{C_{ji_1}, ..., C_{ji_m}\}$ is a set of constraints expressed as intervals or fuzzy sets on $X_j$, $Y_j$ or $Z_j$ variables.

$X_j$ and $Y_j$ variables are defined in the WSDL description of data services. Functionality $\phi_j$ and constraints $C_{ji}$ of a data service $S_{ji}$ are added to the standard WSDL descriptions in the form of annotations. The annotations are represented in the form of SPARQL queries. For instance, the following SPARQL query illustrates the functionality and service constraints of the data service $S_{2,1}$:


#### 3.3 Discovering Relevant Services

Let $Q$ be a preference query. We use an RDF query rewriting algorithm [4] to discover the parts of $Q$ that are covered by each service — recall that in the general case services may match only parts (referred to by $q_1$) of $Q$. The same part $q_1$ is in general covered by one or more services that constitute a class of relevant services and is designated as class $S_1$. A service $S_{ji} \in S_j$ is said to be relevant to a query $Q$ if the functionality of $S_{ji}$ completely matches the part query $q_1$ and its constraints match completely or partially the preference constraints of $q_1$.

As preference constraints of a query are expressed in the rich fuzzy sets framework, their matching degrees with data services constraints may differ from one constraints inclusion method (CIM) to another. Each relevant service is then associated with $|M|$ matching degrees ($m_{1,1}, ..., m_{1,|M|}$) is the set of the used CIMs. For instance, Table 1 shows the matching degrees between each service $S_{ji}$ from Table 1 and its corresponding component $q_1$ (of the query $Q_1$). The degrees are computed by applying the following CIMs:

**Cardinality-based method (CBM) [30].** $\text{Deg}(C \subseteq C^*) = |E^F|/|E|$.

**Implication-based method (IBM) [3].** $\text{Deg}(C \subseteq C^*) = \text{min}_{\mu_1, \mu_2} I(\mu_1(x), \mu_2(x))$ where $I$ stands for a fuzzy implication. The following IBMs are used in our example: Gödel ($I_{Go}$), Lukasiewicz ($I_{Lu}$) and Kleene-Dienes ($I_{Kd}$).

The service $S_{11}$ covering the component $q_1$ does not have a matching degree because there are no constraints imposed by the user on $q_1$. However, each service covering the component $q_2$ is associated with four (the number of methods) degrees. Each matching degree is formulated as a pair of
real values within the range $[0, 1]$, where the first and second values are the matching degrees of the constraints price and warranty, respectively. Similarly, for the matching degrees of the services covering the component $q_i$, the first and second values represent the inclusion degrees of the constraints power and consumption, respectively.

Table 2: Matching degrees between services and fuzzy preferences of $Q_i$

<table>
<thead>
<tr>
<th>$S_{ji}$</th>
<th>$q_j$</th>
<th>$CBM$</th>
<th>$IG_aBM$</th>
<th>$IG_aBM$</th>
<th>$IK_BBM$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{11}$</td>
<td>$q_1$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$S_{21}$</td>
<td>$q_2$</td>
<td>$(1, 0.57)$</td>
<td>$(1, 0)$</td>
<td>$(1, 0)$</td>
<td>$(0.8, 0)$</td>
</tr>
<tr>
<td>$S_{22}$</td>
<td>$q_2$</td>
<td>$(0.89, 1)$</td>
<td>$(0, 1)$</td>
<td>$(0.9, 1)$</td>
<td>$(0.5, 1)$</td>
</tr>
<tr>
<td>$S_{23}$</td>
<td>$q_2$</td>
<td>$(0.2, 0.16)$</td>
<td>$(0, 0)$</td>
<td>$(0, 0)$</td>
<td>$(0, 0)$</td>
</tr>
<tr>
<td>$S_{24}$</td>
<td>$q_2$</td>
<td>$(0.83, 0.88)$</td>
<td>$(0.6, 0.5, 0)$</td>
<td>$(0.6, 0.5, 0)$</td>
<td>$(0.6, 0.5, 0)$</td>
</tr>
<tr>
<td>$S_{31}$</td>
<td>$q_3$</td>
<td>$(0.50, 0.50)$</td>
<td>$(0, 0)$</td>
<td>$(0, 0)$</td>
<td>$(0, 0)$</td>
</tr>
<tr>
<td>$S_{32}$</td>
<td>$q_3$</td>
<td>$(0.79, 0.75)$</td>
<td>$(0, 0.25)$</td>
<td>$(0.6, 0.5, 0)$</td>
<td>$(0.4, 0.5, 0)$</td>
</tr>
<tr>
<td>$S_{33}$</td>
<td>$q_3$</td>
<td>$(0.21, 0.64)$</td>
<td>$(0, 0)$</td>
<td>$(0, 0)$</td>
<td>$(0, 0)$</td>
</tr>
<tr>
<td>$S_{34}$</td>
<td>$q_3$</td>
<td>$(0.83, 0.85)$</td>
<td>$(0.6, 0.5, 0)$</td>
<td>$(0.6, 0.5, 0)$</td>
<td>$(0.6, 0.5, 0)$</td>
</tr>
</tbody>
</table>

3.4 Problem Formulation

Given a query $Q_{<q_1, \ldots, q_n>$ where each $q_i$ is a sub-query (query component). $q_j$ is a tuple $(\overline{q}_j, P_{q_j})$, where $\overline{q}_j$ represents $q_j$ without its preferences $P_{q_j}$. Given a set of services classes $S = \{S_1, \ldots, S_n\}$ where a class $S_j$ regroups the services that are relevant to a query part $q_j$ and a set $M = \{m_1, \ldots, m_M\}$ of CIMs to compute the matching degrees between the constraints of relevant services and the user’s preferences. The problem we are interested in is how to rank data services in each class $S_j$ to select the most relevant ones and how to rank generated data service compositions to select the top-$k$ ones that answer the query $Q$.

4. SERVICES/COMPOSITIONS FUZZY DOMINANCE SCORES

4.1 Fuzzy dominance vs Pareto dominance

Services of the same class $S_i$ have the same functionality, they only differ in terms of constraints, providing thus different matching degrees. Traditional approaches use only one matching criterion to discriminate among relevant services. Most of these approaches aggregate individual matching degrees to compute a global score for each relevant service. One direction is to assign weights to individual matching degrees and, for instance, compute a weighted average of degrees [11]. In so doing, users may not know enough to make trade-offs between different relevancies using numbers (average degrees). Users thus lose the flexibility to select their desired answers by themselves. Computing the skylines from service comes as a natural solution that overcomes this limitation. Skyline computation has received significant consideration in database research [7, 24, 18, 10, 20]. The skyline consists of the set of points which are not dominated by any other point in the sense of Pareto.

Definition 1. (Pareto dominance)

Given two $d$-dimensional points $u$ and $v$. We say that $u$ dominates $v$, denoted by $u \succ v$, iff $u$ is better than or equal to $v$ in all dimensions, and strictly better in at least one dimension, i.e., $\forall i \in [1, d], u_i \geq v_i \land \exists k \in [1, d], u_k > v_k$.

Pareto dominance is not always significant to rank-order points that, however, seem comparable from a user point of view. To illustrate this situation, let $u = (u_1, u_2) = (1, 0)$ and $v = (v_1, v_2) = (0.9, 1)$ be two matching degrees. In Pareto order, we have neither $u \succ v$ nor $v \succ u$, i.e. the instances $u$ and $v$ are incomparable. However, one can consider that $v$ is better than $u$ since $v_2 = 1$ is too much higher than $u_2 = 0$, contrariwise $v_1 = 0.9$ is almost close to $u_1 = 1$. This is why it is interesting to fuzzy the dominance relationship to express the extent to which a matching degree (more or less) dominates another one [6]. In line with the general fuzzification dominance approach discussed in [17], we define below a fuzzy dominance relationship that relies on particular membership function of a graded inequality of the type “strongly larger than”.

Definition 2. (Fuzzy dominance)

Given two $d$-dimensional points $u$ and $v$, we define the fuzzy dominance to express the extent to which $u$ dominates $v$ as:

$$deg(u \succ v) = \frac{\sum_{i=1}^{d} \mu_{\succ}(u_i, v_i)}{d} \tag{1}$$

Where $\mu_{\succ}(u_i, v_i)$ expresses the extent to which $u_i$ is more or less (strongly) greater than $v_i$. $\mu_{\succ}$ can be seen as a monotone membership function defined as:

$$\mu_{\succ}(x, y) = \begin{cases} 0 & \text{if } x - y \leq \varepsilon \\ \frac{x - y - \varepsilon}{\lambda} & \text{otherwise} \end{cases} \tag{2}$$

Where $\lambda > 0$, i.e., $\succ$ is more demanding than the idea of “strictly greater”. We should also have $\varepsilon \geq 0$ in order to ensure that $\succ$ is a relation that agrees with the idea of ‘greater’ in the usual sense.

Let us reconsider the previous instances $u = (1, 0)$, $v = (0.9, 1)$. With $\varepsilon = 0$ and $\lambda = 0.2$, we have $deg(u \succ v) = 0.25$ and $deg(v \succ u) = 0.5$. This is more significant than $|u \succ v| = |v \succ u| = 0$ provided by Pareto dominance, where $|u \succ v| = 1$ if $u \succ v$, 0 otherwise. In the following, we use the defined fuzzy dominance relationship to compute both scores of services and compositions.

4.2 Associating fuzzy score with a service

It is well known that under a single matching method (mono criteria), the dominance relationship is unambiguous. When multiple CIMs are applied (multi-criteria), resulting in different matching degrees for the same couple of constraints, the dominance relationship becomes uncertain. The model proposed in [21], namely probabilistic skyline overcomes this problem. Contrariwise, Skoutas et al. show in [22, 23] the limitations of the probabilistic skyline to rank Web services and introduce the Pareto dominating score of individual Web services. We generalize this score to fuzzy dominance and propose the fuzzy dominating score (FDS). An FDS of a service $S_{ji}$ indicates the average extent to which $S_{ji}$ dominates the whole services of its class $S_j$.

Definition 3. (Fuzzy dominating score of a service)

The fuzzy dominating score (FDS) of a service $S_{ji}$ in its class $S_j$ is defined as:

$$FDS(S_{ji}) = \frac{1}{(|S_j| - 1)|M|} \sum_{h=1}^{M} \sum_{S_{jk} \in S_j, r=1}^{M} \sum_{S_{jk} \neq S_{ji}} \sum_{S_{jk} \neq S_{ji}} deg(S_{ji}^{h} \succ S_{jk}^{r}) \tag{3}$$
where $S_{i}^{h}$ is the matching degree of the service $S_{i}$ obtained by applying the $h^{th}$ CIM. The term $(|S_{i}| - 1)$ is used to normalize the FDS score and make it in the range $[0,1]$. Table 3 shows the fuzzy dominating scores of the data Web services in our running example.

### 4.3 Associating fuzzy score with a composition

Different data service compositions can be generated from different $S_{i}$ service classes to answer a user query. To rank such generated compositions, we extend the previous FDS definition to service composition and associate each composition with an FDS. The FDS of a composition $C$ is an aggregation of different FDSs of its component services.

Let $C = \{S_{i1}, ..., S_{ini}\}$ be a composition of $n$ services. Let also $d = d_{1} + ... + d_{n}$ be the number of user preference constraints where $d_{j}$ is the number of constraints involved in the service $S_{ij}$. The FDS of $C$ is then computed as follows:

$$FDS(C) = \frac{1}{d} \sum_{j=1}^{n} d_{j} \cdot FDS(S_{ij})$$  \hspace{1cm} (4)

It is important to note that not all compositions are valid. A composition $C$ of data Web services is valid if (i) it covers the user query $Q$, (ii) it contains one and only one service from each service class $S_{i}$ and (iii) it is executable. A composition is said to be executable if all input parameters necessary for the invocation of its component services are bound. For more details see [4].

**Table 3: Services’ scores and services**

<table>
<thead>
<tr>
<th>Services</th>
<th>Class</th>
<th>Score</th>
<th>Top-k</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{i1}$</td>
<td>$S_{1}$</td>
<td>-</td>
<td>$S_{i1}$</td>
</tr>
<tr>
<td>$S_{i2}$</td>
<td>$S_{2}$</td>
<td>0.487</td>
<td>$S_{i2}$</td>
</tr>
<tr>
<td>$S_{i3}$</td>
<td>$S_{2}$</td>
<td>0.653</td>
<td>$S_{i3}$</td>
</tr>
<tr>
<td>$S_{i4}$</td>
<td>$S_{3}$</td>
<td>0.538</td>
<td>$S_{i4}$</td>
</tr>
<tr>
<td>$S_{i5}$</td>
<td>$S_{3}$</td>
<td>0.094</td>
<td>$S_{i5}$</td>
</tr>
<tr>
<td>$S_{i6}$</td>
<td>$S_{3}$</td>
<td>0.393</td>
<td>$S_{i6}$</td>
</tr>
<tr>
<td>$S_{i7}$</td>
<td>$S_{3}$</td>
<td>0.130</td>
<td>$S_{i7}$</td>
</tr>
<tr>
<td>$S_{i8}$</td>
<td>$S_{3}$</td>
<td>0.743</td>
<td>$S_{i8}$</td>
</tr>
</tbody>
</table>

### 5. TOP-K DATA SERVICE COMPOSITIONS

#### 5.1 An Efficient Generation Technique

A straightforward method to find the top-$k$ data Web service compositions that answer a query is to generate all possible compositions, compute their scores, and return the top-$k$ ones. However, this approach results in a high computational cost, as it needs to generate all possible compositions, whereas, most of them are not in the top-$k$. In the following, we provide an optimization technique to find the top-$k$ data Web service compositions. This technique allows eliminating relevant services $S_{ij}$ from their classes $S_{i}$ before generating the compositions, i.e., services that are sure that if they are composed with other ones, the obtained compositions are not in the top-$k$. The idea is: we first compute the score of each service in its class, then only the best services in each class are retained, after that we compose the retained services, finally, we compute the score of the obtained compositions and return the top-$k$ ones. To this end, we introduce the following lemma and theorem.

**Lemma 1.** Let $C = \{S_{i1}, ..., S_{ini}, S\}$ and $C' = \{S_{i1}, ..., S_{ini}, S'\}$ be two similar service compositions that only differ in the services $S$ and $S'$. Then, the following statement holds: $FDS(S) > FDS(S') \implies FDS(C) > FDS(C')$.

**Proof.** Denoting by $d'$ the number of constraints involved in $S$ and $S'$, we have:

$$FDS(C) = \frac{1}{d} \sum_{j=1}^{n} d_{j} \cdot FDS(s_{ij}) + d' \cdot FDS(S)$$

and

$$FDS(C') = \frac{1}{d} \sum_{j=1}^{n} d_{j} \cdot DSF(s_{ij}) + d' \cdot FDS(S').$$

Then, $FDS(C) - FDS(C') = d' (FDS(S) - FDS(S'))$. Since $d' > 0$ and $FDS(S) - FDS(S') > 0$, we have $FDS(C) > FDS(C')$. \hfill $\Box$

Lemma 1 means that the best services in their classes will generate the best compositions.

**Theorem 1.** Let $C = \{S_{i1}, ..., S_{ini}\}$ be a service composition and $top-k(S_{i})$ (resp. $top-k(S_{i})'$) be the top-$k$ services of the class $S_{i}$ (resp. the top-$k$ compositions). Then, $\exists S_{ij} \in C; S_{ij} \notin top-k(S_{i}) \implies C \notin top-k(C)$.

**Proof.** Assume that $C \in top-k(C') \land \exists S_{ij} \in C; S_{ij} \notin top-k(S_{i})$. This means that $\exists S_{ij1}, ..., S_{ijn} \in S_{i}; FDS(S_{ij1}) > FDS(S_{ij}). Now, by replacing $S_{ij}$ in $C$ with the services $S_{ij1}, ..., S_{ijn}$ we obtain $k$ compositions $C_{1}, ..., C_{k}$ such as $FDS(C_{i}) > FDS(C)$ according to the Lemma 1. This contradicts our hypothesis. Hence, $C \notin top-k(C)$. \hfill $\Box$

Theorem 1 means that the top-$k$ sets of the different service classes are sufficient to compute the top-$k$ data Web service compositions that answer the considered query.

The fourth column of Table 3 shows the top-$k$ (where $k = 2$) data services in each service class using the FDS scores. Thus, relevant data services that are not in the top-$k$ of their classes are eliminated. They are crossed out in Table 3. The other data services are retained. The top-$k$ data service compositions are generated from the different top-$k$ $S_{i}$ classes. Table 4 shows the possible compositions along with their fuzzy dominating scores and the top-$k$ ones of our running example.

**Table 4: Compositions’ scores and top-k ones**

<table>
<thead>
<tr>
<th>Compositions</th>
<th>Score</th>
<th>Top-k</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{1} = {S_{i1}, S_{i2}, S_{i3}}$</td>
<td>0.623</td>
<td>$C_{1}$</td>
</tr>
<tr>
<td>$C_{2} = {S_{i1}, S_{i2}, S_{i4}}$</td>
<td>0.698</td>
<td>$C_{2}$</td>
</tr>
<tr>
<td>$C_{3} = {S_{i1}, S_{i4}, S_{i3}}$</td>
<td>0.506</td>
<td>$C_{3}$</td>
</tr>
<tr>
<td>$C_{4} = {S_{i1}, S_{i4}, S_{i4}}$</td>
<td>0.640</td>
<td>$C_{4}$</td>
</tr>
</tbody>
</table>

### 5.2 Algorithm

The algorithm, hereafter referred as TKDSC, computes the top-$k$ data service compositions according to the fuzzy dominating scores. The algorithm proceeds as following.

**Step.1 computing the matching degrees (lines 1-13).** Each service class whose services cover a query component is added to the list of relevant classes. If its services touch the query’s user preferences, we compute its different matching degrees according to the number of methods.

**Step.2 eliminating less relevant services (lines 14-23).** For each class whose services do not touch the user preferences, we select randomly $k$ services since they are all equal
Algorithm 1 TKDSC

Require: $Q$ a preference query; $S = \{S_1, ..., S_n\}$ a set of service classes; $M = \{m_1, ..., m_{|M|}\}$ a set of methods; $k \in \mathbb{N}; \varepsilon > 0; \lambda > 0$;

Ensure: the top-$k$ service compositions
1: for all $S_j$ in $S$ do
2: \hspace{1em} $S \leftarrow random(S_j, 1)$;
3: \hspace{1em} if $\exists q_i \in Q; cover(S_j, q_i)$ then
4: \hspace{2em} $R \leftarrow R \cup S_j$;
5: \hspace{1em} if $P_{q_j} \neq \emptyset$ then
6: \hspace{2em} \hspace{1em} for all $S_{ij}$ in $S_j$ do
7: \hspace{3em} \hspace{1em} for all $m$ in $M$ do
8: \hspace{4em} \hspace{1em} ComputeMatchingDegree($C_{ji}, P_{q_j}, m$);
9: \hspace{2em} \hspace{1em} end for
10: \hspace{1em} \hspace{1em} end for
11: \hspace{1em} end if
12: end if
13: end for
14: for all $S_j$ in $R$ do
15: \hspace{1em} if $P_{q_j} = \emptyset$ then
16: \hspace{2em} $top-k; S_j \leftarrow random(S_j, k)$;
17: \hspace{1em} else
18: \hspace{2em} \hspace{1em} for all $S_{ij}$ in $S_j$ do
19: \hspace{3em} \hspace{1em} ComputeScore($S_{ij}$);
20: \hspace{2em} \hspace{1em} end for
21: \hspace{2em} $top-k; S_j \leftarrow top(k, S_j)$;
22: \hspace{1em} end if
23: \hspace{1em} end for
24: $C \leftarrow ComposeServices(top-k; S_{j1}, ..., top-k; S_{jm})$;
25: for all $C$ in $C$ do
26: \hspace{1em} ComputeCompositionScore($C$);
27: end for
28: return $top(k, C)$;

with respect to user preferences. Otherwise (i.e., its services touch the user preferences), we first compute the score of its services and then retain only the top-$k$ ones.

Step 3 returning top-$k$ compositions (lines 24–28).
First, we compose services from only the retained ones, i.e., the top-$k$ in each class. Then we compute the score of generated compositions and finally we provide the user with the top-$k$ ones.

6. SYSTEM ARCHITECTURE AND EXPERIMENTAL EVALUATION

6.1 System Architecture

Figure 2 presents our implemented top-$k$ data service compositions system. The system consists of the three major modules: Annotation Module, Interactive Query Formulation Module and Top-$k$ Service Compositions Module.

The Annotation Module allows service providers to annotate WSDL description files of services with fuzzy sets to represent linguistic terms and with SPARQL queries to represent the functionality and constraints of services. This annotation is implemented by adding a new element called “rdfQuery” to the XML Schema of WSDL as in WSDL-S approach. The annotated WSDL files are then published on a service registry. The ontology manager uses Jena API to manage domain ontology, i.e., to add/delete concepts.

The Interactive Query Formulation Module provides users with a GUI implemented with Java Swing to interactively formulate their queries over a domain ontology. Users are not required to know any specific ontology query languages to express their queries.

The Top-$k$ Service Compositions Module consists of five components. The RDF Query Rewriter implements an efficient RDF query rewriting algorithm (RDF Query Rewriter) to identify the relevant services that match (some parts of) a user query. For that purpose, it exploits the functionalities in the service description files. The Service Locator feeds the Query Rewriter with services that most likely match a given query. The Top-$K$ Compositions component computes (i) the matching degrees of relevant services, (ii) the fuzzy dominating scores of relevant services, (iii) the top-$k$ services of each relevant service class and (iv) the fuzzy compositions scores to return the top-$k$ compositions. The top-$k$ compositions are then translated by the composition plan generator into execution plans expressed in the XPDL language. They are executed by a workflow execution engine; we use the Sarasvati execution engine from Google.

6.2 Experimental Evaluation

Our objective is to prove the efficiency and the scalability of our proposed approach for top-$k$ data Web service compositions. For this purpose, we implemented a top-$k$ service generator. The generator takes as input a set of (real-life) model services (each representing a class of services) and their associated fuzzy constraints and produces for each model service a set of synthetic Web services and their associated synthetic fuzzy constraints. In the experiments we evaluated the effects of the following parameters: (i) the number of services per class, (ii) the service classes number, (iii) the number of fuzzy constraints per class, (iv) the number of matching methods and (v) the parameter $k$. The default values of these parameters are: 400, 4, 4, 5, 5, respectively.

The algorithm (i.e., TKDSC) is implemented in Java. The experiments were conducted on a 2.00 GHz Intel dual core CPU and 2 GB of RAM, running Windows. The results of the experiments are presented in Figure 3.

6.2.1 Performance vs. number of services per class

We measured the average execution time required to solve the top-$k$ service compositions problem as the number of services per class increases, varying the number of services
per class from 100 to 1000. The results of this experiment are presented in Figure 3 (plot-a). The results show that our framework can handle hundreds of services per class in a reasonable time.

6.2.2 Performance vs number of classes

We measured the average execution time required to solve the top-k service compositions problem as the number of service classes increases. We varied the classes number from 1 to 6. The results of this experiment in Figure 3 (plot-b) show that the execution time is proportional to the number of service classes.

6.2.3 Performance vs number of constraints per service

We varied the fuzzy constraints number from 1 to 10 and measured the average execution time required to compute the top-k service compositions. Figure 3 (plot-c) shows the time required to compute the top-k service compositions.

6.2.4 Performance vs. number of matching methods

We varied the number of matching methods from 1 to 10. We measured the average execution time required to compute the top-k service compositions. The results of this experiment are shown in Figure 3 (plot-d).

6.2.5 Performance vs. k

We measured the average execution time required to compute the top-k service compositions as the value of k increases. We varied the value of k from 3 to 5. The results of this experiment in Figure 3 (plot-e) show that the execution time increases as the value of k increases.

7. RELATED WORK

Preferences in Web service selection/composition have received much attention in the service computing community during the last years. Taking user preferences into account allows to rank candidate services/compositions and return only the best ones to the user. Hereafter, we review some works for ranking and selecting Web services.

ServiceTrust [15] calculates reputations of services from users. It introduces transactional trust to detect QoS abuse, where malicious services gain reputation from small transactions and cheat at large ones. However, ServiceTrust models transactions as binary events (success or failure) and combines reports from users without taking their preferences into account. In [26], the authors use a qualitative graphical representation of preference, CP-nets, to deal with services selection in terms of user preferences. This approach can reason about a user’s incomplete and constrained preference. In [19], a method to rank semantic web services is proposed. It is based on computing the matching degree between a set of requested NFPs (Non-Functional Properties) and a set of NFPs offered by the discovered Web services. NFPs cover QoS aspects, but also other business-related properties such as pricing and insurance. Semantic annotations are used for describing NFPs and the ranking process is achieved by using some automatic reasoning techniques that exploit the annotations. However, the problem of composition is not addressed in these works.

Agarwal and Lamparter [1] propose an automated Web service selection approach for composition. Web service combinations can be compared and ranked according to user preferences. Preferences are modeled as a set of fuzzy IF-THEN rules. The IF part contains fuzzy descriptions of the various properties of a service (i.e., a concrete Web service composition) and the THEN part is one of the fuzzy characterizations of a special concept called Rank. A fuzzy rule describes which combination of attribute values a user is willing to accept to which degree, where attribute values and degrees of acceptance are defined in a fuzzy way. ServiceRank [27] considers the QoS aspects as well as the social perspectives of services. Services that have good QoS and are frequently invoked by others are more trusted by the community and will be assigned high ranks. In [25], the authors propose a system for conducting qualitative Web service selection in the presence of incomplete or conflicting user preferences. The paradigm of CP-nets is used to model user preferences. The system utilizes the history of users to amend the preferences of active users, thus improving the results of service selection.

The work most related to ours is [22, 23], where the authors consider dominance relationships between Web services based on their degrees of match to a given request in order to rank available services. Distinct scores based on the notion of dominance are defined for assessing when a service is objectively interesting. However, that work only considers selection of single services, without dealing with the problem of composition nor the user preferences.

Recent approaches, focus on computing the skyline from Web services. All these approaches focus on selecting Web services based on QoS parameters. The work in [2] focuses on the selection of skyline services for QoS based Web service composition. A method for determining which QoS levels of a service should be improved so that it is not dominated by other services is also discussed. In [29], the authors propose a skyline computation approach for service selection. The resulting skyline, called multi-service skyline, enables services users to optimally and efficiently access sets of service as an integrated service package. In the robust work [28], Yu and Bouguettaya address the problem of uncertainty inherent in QoS and compute the skylines from service providers. A service skyline can be regarded as a set of service providers that are not dominated by others in terms of QoS aspects that interest all users. To this end, a concept called p-dominant service skyline is defined. A provider S belongs to the p-dominant skyline if the probability that S is dominated by any other provider is less than p. The authors provide also a discussion about the interest of p-dominant skyline w.r.t.
the notion of p-skyline proposed in [21]. In [5], we propose a new concept called α-dominant service skyline based on a fuzzy dominance relationship to address the majors issues of the traditional service skyline, i.e., privileging Web services with a bad compromise between QoS attributes and not allowing users to control the size of the returned set of Web services. However, these works do not take user preferences into account and except for [2] the problem of composition is not addressed.

8. CONCLUSION

In this paper, we addressed the problem of top-k retrieval of data Web service compositions to answer fuzzy preference queries under different matching methods. We presented a suitable ranking criterion based on a fuzzification of Pareto dominance and developed a suitable algorithm for computing the top-k data Web service compositions. Our experimental evaluation shows that our approach can retrieve the quality of the results and combine this work with QoS aspect.

9. REFERENCES

About the authors:

Karim Benouaret received the Master degree in computer science from the University of Bejaia (Algeria) in 2009. He is currently a PhD student at the University of Lyon.

His current research interests lie in the area of service computing, in particular query optimization over Web services selection and composition (e.g., skyline query, top-k query, etc.). His works were published in a well-known conference proceeding (VLDB 2011 demo paper, IEEE ICWS 2011, IEEE SCC 2011 and ACM SAC 2011).

Djamal Benslimane (http://www710.univ-lyon1.fr/~dbenslim/) is currently a Professor at Lyon 1 University in France. He is the head of the computer science department at the IUT Lyon 1. His research interests lie in the areas of Services computing and database interoperability. His recent works were published in IEEE Internet Computing, IEEE Transactions on Services Computing, IEEE Transactions on Systems, Man, and Cybernetics, Communications of the ACM, ACM Transactions on Internet Computing, ACM Transactions on Software Engineering and Methodology.

Allel Hadjali received the Engineer and "Magister" degrees (1988 and 1991) from the University of Tizi-ouzou (Algeria). From 2001 to 2002, He has prepared the Ph.D. degree at the University of Toulouse 3 (France) and has obtained the "Doctorat d'Etat" degree from the University of Tizi-ouzou in 2003. He joined the University of Rennes 1 (France) in 2003. He is currently an Associate Professor.

A Hough-transform-based Anomaly Detector with an Adaptive Time Interval

Romain Fontugne
The Graduate University for Advanced Studies
Tokyo, Japan

Kensuke Fukuda
National Institute of Informatics / PRESTO JST
Tokyo, Japan

ABSTRACT

Internet traffic anomalies are a serious problem that compromises the availability of optimal network resources. Numerous anomaly detectors have recently been proposed, but maintaining their parameters optimally tuned is a difficult task that discredits their effectiveness for daily usage. This article proposes a new anomaly detection method based on pattern recognition and investigates the relationship between its parameter set and the traffic characteristics. This analysis highlights that constantly achieving a high detection rate requires continuous adjustments to the parameters according to the traffic fluctuations. Therefore, an adaptive time interval mechanism is proposed to enhance the robustness of the detection method to traffic variations. This adaptive anomaly detection method is evaluated by comparing it to three other anomaly detectors using four years of real backbone traffic. The evaluation reveals that the proposed adaptive detection method outperforms the other methods in terms of the true positive and false positive rate.

Categories and Subject Descriptors
C.2.3 [Computer-Communication Networks]: Network Operations—Network monitoring

General Terms
Measurement, Security, Performance

Keywords
Internet traffic, Anomaly detection, Pattern recognition

1. INTRODUCTION

The success of Internet services results in a constant network traffic growth along with an increasing number of anomalies such as remote attacks and misconfigurations. These anomalies represent a large fraction of the Internet traffic that is unwanted and penalizes legitimate users from accessing optimal network resources. Therefore, detecting and diagnosing these threats are crucial tasks for network operators that are trying to maintain the Internet resources made available. Intensive studies have been carried out in this field, but the proposed anomaly detection methods still have common drawbacks [15, 12]. Indeed, the sensitivity of adapting the parameter set of these methods to traffic variations are still open issues. Since the relationship between the parameter setting and traffic characteristics is misunderstood, in practice, selecting the optimal parameters is challenging.

Only a few works have investigated this drawback currently discrediting anomaly detectors. A careful study of the detectors based on principal component analysis (PCA) was carried out by Ringberg et al. [15], and they identified four main challenges including the sensitivity to analyzed traffic and parameter tuning. In addition, an attempt to automatically tune a method based on gamma modeling and sketches was conducted by Himura et al. [12]. They designed a learning process for predicting the optimal parameters regarding the best parameters for past data. However, this method suffers from a high error rate as unexpected events do appear.

Recently, a pattern recognition based method has been proposed [10, 11]. The main idea of this detection method is to monitor the traffic in 2D pictures where anomalies appear as “lines”, which are easily identifiable using a pattern recognition technique called the Hough transform [7]. One advantage of this method is that its simple principles allow us to intuitively select a suitable parameter set. The optimal values of the parameters, however, fluctuate along with the traffic throughput variations and require continuous adjustments, making it impractical for real usage.

In order to provide a detector that is easily tunable and robust to traffic variations, this article follows a similar approach to [10], but it uses fundamentally different 2D pictures that allow for better highlighting anomalies. Moreover, the main contribution of this work is to obtain a complete understanding of the proposed method parameter set and provide a mechanism that automatically tunes it based on the traffic variations. The advantages of this adaptive method are demonstrated by comparing its results to those obtained using fixed parameter tunings and those of three other anomaly detectors using four years of real Internet traffic. The results highlight the superiority of the proposed method in terms of the true positive and false positive rates,
Table 1: Different kinds of common anomalies and their particular traffic feature distributions.

<table>
<thead>
<tr>
<th>Anomaly</th>
<th>Traffic feature distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port scan</td>
<td>Traffic distributed in destination port space and concentrated on single destination host.</td>
</tr>
<tr>
<td>Network scan, Worm, Exploit</td>
<td>Traffic distributed in destination address space and concentrated on limited number of destination ports.</td>
</tr>
<tr>
<td>DDoS, Netbot, Flash crowd</td>
<td>Traffic distributed in source address space and concentrated on limited number of destination addresses.</td>
</tr>
</tbody>
</table>

emphasizing that automatically adjusting the parameter set in regard to the traffic fluctuations is crucial for continuously performing an accurate level of detection. Inspecting the false negative rate of the proposed method allows us to describe the particular class of anomaly that is inherently missed by the proposed detector. Thus, the shortcomings of the detector are well-defined and complementary detectors are suggested.

2. ABNORMAL DISTRIBUTION OF TRAFFIC FEATURES

Recent works have identified anomalous traffic as alterations in the distributions of the traffic features [14, 10, 4, 19, 6]. For example, Table 1 lists several kinds of anomalies commonly identified in Internet traffic. Each kind of anomaly inherently affects the distribution of two traffic features. Similarly, in this article an anomaly refers to a set of flows altering the distribution of at least one of the four following traffic features: the source IP address, destination IP address, source port, and destination port. However, the proposed approach for observing these alterations in the traffic feature distributions is substantially different from that in other works. Previously, anomalies have been mainly detected by identifying the outliers in the aggregated traffic using different formalisms — e.g., signals [14], histograms [6, 4], or matrices [18] — whereas, the proposed method identifies particular patterns in pictures. The analyzed pictures are two-dimensional scatter plots, where each axis represents a traffic feature, each plot stands for traffic flows, and the particular traffic feature distributions of the anomalies are easily identifiable as lines.

Figure 1 shows two examples of the pictures analyzed in this article. Figure 1a displays traffic with regard to its destination port and destination address. This graphical representation of the traffic makes it easy to discriminate the port scan, network scan, worm, and exploit from the benign traffic as they appear as lines in the picture (Fig. 1a). Figure 1b, however, displays the traffic in regard to its destination and source addresses, and permits other kinds of anomaly to be observed. For instance, distributed denial of service (DDoS), flash crowd and botnet activities appear as horizontal lines in this scatter plot.

The three main advantages of this approach over the previous works are [15]: (1) the anomalous flows are inherently pinpointed in the scatter plots, whereas the identification of the anomalous flows detected in a signal requires additional extraction mechanisms [4, 16]. (2) The proposed approach is able to monitor the pattern of a large-scale anomaly whereas the methods detecting anomalous traffic as outliers fail if a majority of the traffic is contaminated by anomalies (e.g., outbreak of virus). (3) In regard to the traffic features monitored by the pictures and the direction of the identified line, one can easily deduce the kind of observed anomaly.

3. ANOMALY DETECTION METHOD

The anomaly detection method proposed in this article consists of five main steps:

1. The traffic of the current time interval is mapped onto five different pictures.
2. The Hough transform is computed on each picture to uncover the plot distributions.
3. Abnormal plot distributions are detected in the Hough spaces.
4. Traffic information corresponding to the anomalous plots are retrieved and reported.
5. The time interval is shifted and step 1 is repeated.

3.1 Pictures computation

The proposed approach takes advantage of several kinds of pictures to monitor the different aspects of the traffic and highlight the different kinds of anomalies. The analyzed pictures are 2-D scatter plots designed from four traffic features: {source IP address, destination IP address, source port, destination port}. For the remainder of this paper the term traffic features will refer to only these four traffic features. The five picture categories correspond to all the possible pairs of traffic features containing IP address. Namely, the x and y axis of the picture, respectively, correspond to the following pairs of features:

- Source IP address, destination IP address
- Source IP address, source port
- Source IP address, destination port
- Destination IP address, source port
- Destination IP address, destination port

Applied Computing Review 42
A flow in the analyzed pictures is represented by a plot that is located using the two following mechanisms. (1) The port space is shrunk to the size of the pictures: Let assume a 1000-pixel picture that has a y axis standing for the source port, then a http flow, i.e., $SrcPort = 80$, is plotted at $y = [SrcPort \times ySize/2^{16}] = [80 \times 1000/65535] = 1$, and each pixel of the picture represents approximately $65535/1000 = 65$ distinct port numbers. (2) The IP address space is at first hashed by ignoring the first $h$ bits of the addresses and then shrunk to the size of the picture. For example, supposing $h = 16$ and a 1000 pixel wide picture ($xSize = 1000$) with an $x$ axis as the source IP, then a flow from the source IP 192.168.10.10 is plotted at $x = ((SrcIP \mod 2^{32-h}) \times xSize/2^{32-h}) = (192.168.10.10 \mod 2^{16})\times1000/2^{16} = (0.0.10.10)\times1000/2^{16} = 39$. Notice that this article only deals with square pictures, commonly used to detect the parametric structures (e.g., elements of the Hough space that are higher than 3σ).

### 3.2 Hough transform

A well-known image processing technique called the Hough transform [7, 11] helps us in extracting the relevant information from computed pictures. The Hough transform is commonly used to detect the parametric structures (e.g., line, circle, or ellipse) in pictures and has the advantage of being robust to noise and able to detect incomplete shapes.

The basic usage of the Hough transform allows for the identification of lines in a picture. It consists of a voting procedure, where each plot of the picture votes for the lines it belongs to. Formally, each point in the picture $p = (x_p, y_p)$ votes for all the $\theta$ and $\rho$ that satisfy $p = x_p \cdot \cos(\theta) + y_p \cdot \sin(\theta)$ (line equation in polar coordinates). All the votes are stored in a two-dimensional array, called the Hough space, in which one dimension stands for $\theta$ and one for $\rho$. Figure 2 depicts an example of the Hough transform. The analyzed picture (Fig.2a) contains three plots, and the votes for each plot are represented by a curve in the Hough space (Fig.2b). The maximum number of votes in the Hough space is obviously at the intersection of the three curves $I = (\theta_0, \rho_0)$, identifying the line passing through the 3 plots, $\rho_0 = x \cdot \cos(\theta_0) + y \cdot \sin(\theta_0)$.

In order to find the local maxima in the Hough space, thus the prominent lines in the picture, a robust peak detection based on the standard deviation $\sigma$ of the Hough space is implemented. Therefore, all flows corresponding to the elements of the Hough space that are higher than $3\sigma$ are reported as anomalous.

### 3.3 Complexity

The computational complexity of the proposed method is mainly one of the Hough transforms that is linear to the number of plots in picture. In a worst case scenario, each plot represents a single flow so the number of plots in the pictures is equal to the total number of flows $N$. Let $f = 5$ be the number of picture categories, $t$ the traffic duration divided by the time interval, and $n_{i,j,k}$ the number of plots in the picture $k$ of category $i$ at the time interval $j$. The cost of the proposed algorithm in the worst case is linear to $N$:

$$\sum_{i=1}^{t} \sum_{j=1}^{t} O(n_{i,j}) = 5 \sum_{t=1}^{t} O(N) = O(N)$$

In our experiments, the proposed method takes about one minute to analyze a 15-minute traffic trace from the MAWI archive.

### 4. DATA AND PROCESSING

All the experiments conducted in this work are based on the traffic traces publicly available in the MAWI archive [5]. This database provides daily backbone traffic traces that contain 15-minutes of traffic taken from a trans-Pacific link between Japan and the U.S. Since 2001 this link was an 18 Mbps CAR on a 100 Mbps link, but it was replaced by a full 100 Mbps link in 2006/07/01. This article particularly focuses on two data sets from the MAWI archive; (1) the first week of August 2004 was particularly affected by the Sasser worm [3, 10] and provides valuable support for illustrating the benefits of the proposed method. (2) All the traffic recorded from 2003 to 2006 allowed us to evaluate the global performance of the proposed method by comparing its results to the ones of other anomaly detectors.

Due to the lack of ground truth data for backbone traffic, the evaluation of the proposed detector relies on heuristics that is fundamentally independent from the principle of the proposed method (Table 2). Indeed, these heuristics is based on well-known port numbers and abnormal usages of TCP flags [3, 10], whereas the proposed method uses only the port numbers as indexes and does not rely on the application information related to them nor the TCP flags. Heuristics classifies traffic into two categories, attack and special.
and helps in quantifying the effectiveness of the detection method.

An anomaly detector is expected to report more traffic classified as attacks than those labeled special. Thus, the accuracy of a detector is defined as the ratio of the alarms classified as attacks by the heuristics listed in Table 2.

5. PARAMETER TUNING AND DRAWBACKS

5.1 Experimental parameter tuning

The following experiments aim at finding the optimal parameter tuning of the proposed method using one week of traffic affected by the Sasser worm (Section 4). Furthermore, these experiments uncover the correlation between the two main parameters, i.e., the size of picture and the time interval, and show that the performances of the proposed method are not affected by any variance in the $h$ value as long as the number of possible indexes is higher than the picture size, $2^{32-h} > x\text{Size}$.

Figure 3 depicts the average accuracy of the detection method using numerous parameter values. It highlights that the proposed method is able to achieve an accuracy that is higher than 0.9 for any time interval > 4s and a suitable picture size. Furthermore, Fig. 3 indicates that the optimal picture size is proportional to the size of the time interval. For instance, if the time interval is less than 8s the best performance is obtained with a picture size set to 1024, whereas the time interval ranges (9, 16) are suitable for a picture size equal to 2048, and so forth. Intuitively, a larger time interval involves a greater number of plots in the pictures; thus, to avoid meaningless saturated pictures, the optimal size of a picture increases along with the size of the time interval.

Although the specific values given here are suitable for the analyzed traffic, different values might be more effective for traffic having different properties. Obviously, traffic with the same properties but a higher throughput displays more plots in the pictures, and thus in this case, smaller time intervals are required to maintain an acceptable number of plots in the pictures.

5.2 Evaluation of optimal parameter

The time interval is the parameter that controls the amount of traffic displayed in the pictures. Thus, as the proposed method inherently translates the traffic flows to the plots in the pictures, the time interval allows us to select the quantity of plots appearing in the pictures. The challenge in setting the time interval is the trade-off between displaying enough plots to have relevant pictures and limiting the surrounding noise representing the legitimate traffic and hiding anomalies.

The sensitivity of the implemented Hough transform to the number of plots in the pictures is analyzed using synthetic pictures that have a random line and various amounts of uniformly distributed noise. The algorithm was performed 100 times on different pictures with the same level of noise. If the 100 tests are successful then the noise is increased and the algorithm is again performed. The highest noise level for which all 100 executions of the algorithm succeed defines the maximum acceptable number of plots in a picture. This experiment was conducted using six different picture sizes, as indicated in Fig. 4a. As expected, the maximum acceptable number of plots in the pictures increases with the picture size. Figure 4a shows that the maximum acceptable number of plots for picture sizes of 1024, 2048, 4096, and 8192 are respectively 33000, 95000, 275000, and 781000. Figure 4b shows that this increase is not linear to the area of the picture and the common upper bound for all the considered picture sizes is approximately 1% of the picture area.
Figure 5: Plot growth for different picture categories ($xSize = 8192$).

### 5.3 Dispersion of plots in pictures

The previous section provided an insight on how to select the suitable time interval for a particular picture, but the proposed method analyzes five different pictures at the same time. A crucial task is to understand the divergence between the different kinds of pictures. Since the five picture categories monitor distinct feature spaces, plots corresponding to the same traffic are differently dispersed in all the pictures. Therefore, the traffic is usually depicted by using a different number of plots for two pictures from different categories. For example, Fig. 5a shows the number of plots for the five kinds of pictures for several time interval sizes. This figure highlights that the number of plots appearing in each picture category increases at different rates. A slow increase in the number of plots means that many flows share the same instance in the monitored feature spaces, whereas a rapid growth highlights the flows spreading into the observed feature spaces. The rate of increase of the plots for each picture category is strongly related to the throughput and the dispersion of the traffic in the feature space.

Since anomalies alter the traffic feature distribution, they also significantly affect the increase in the number of plots. Figure 5b is a typical example where the increase in plots for certain picture categories is rapidly increasing due to anomalous traffic. Indeed, the traffic analyzed in Fig. 5b contains an outbreak of the Sasser worm highlighting a considerable increase in the number of plots for two picture categories monitoring the destination address. This observation is in accord with the behavior of the Sasser worm manually observed in the traffic trace, that is, the worm tries to infect numerous remote hosts to spread throughout the network.

Despite their differences, the two traffic analyzed in Fig. 5 are taken from the same traffic trace (Fig. 5b representing the first three minutes of the traffic trace, whereas Fig. 5a is the traffic recorded three minutes later), illustrating two drawbacks of the proposed method. (1) For the same traffic, the number of plots in all the picture categories is significantly different. Thus, the suitable time interval for a picture from a certain category does not necessarily suit the pictures from the other categories. (2) The increase in plots for a certain picture category sharply varies especially when anomalous traffic appears. Thus, the suitable time interval for a single picture category fluctuates over time.

### 6. ADAPTIVE TIME INTERVAL

Here, an improved version of the anomaly detection method is proposed to overcome the drawback identified in the previous section. This new version assigns different time intervals to all the picture categories and adapts these time intervals to the traffic variation. Therefore, the value of the time intervals is no longer a fixed value taken as an input, but it is automatically computed by taking into account the throughput and the traffic distribution in the traffic feature spaces.

The proposed improvement consists of controlling the amount of monitored traffic based on the quantity of plots in the picture instead of the time interval. The Hough transform is performed only if a certain number of plots are displayed in the picture (regardless of the time interval corresponding to the traffic mapped into the picture), and other pictures keep monitoring the traffic until they display a sufficient number of plots. Therefore, all the pictures stand for different time intervals and the Hough transform is performed at different instants of time for each picture. The first two steps of the algorithm proposed in Section 3 are replaced by:

1. Map traffic to pictures until a picture displays $p$ plots.
2. Compute the Hough transform for pictures with $p$ plots.

In addition, the time interval parameter is replaced by $p$, which is the number of plots required to perform the Hough transform. The value of $p$ is directly deduced from the picture size to assure the success of the Hough transform. The upper bound for $p$ is 1% of the picture area (Section 5.2), and the lower values help in quickly reporting the anomalies since the Hough transform is performed earlier. However, too small $p$ values result in irrelevant pictures as the sample traffic displayed in pictures is insignificant. In the following experiments, $p$ is arbitrarily set to 0.5% of the picture area, $p = 0.05 \cdot xSize^2$. Hereafter, this new version of the detection method is referred to as the adaptive method.

### 6.1 Performance improvement

The benefit of the adaptive method is evaluated by using one week of traffic (Section 4). For clarity reasons and because all the traffic traces reach a similar conclusion, the following focuses only on the first day of the analyzed traffic.

#### 6.1.1 Robustness to traffic variation

Figure 6 displays the time intervals corresponding to all the pictures computed during the analysis of the 15 minutes of traffic. The first four minutes of this traffic are significantly affected by the Sasser worm resulting in a higher
throughput and an increase in the number of destination addresses. Nevertheless, the method successfully handled the traffic variation, that is, the time intervals represented by the pictures monitoring the destination address remain from 1 to 5 seconds during the Sasser outbreak (Fig. 6). However, the same quantities range from 14 to 25 seconds during the last four minutes of traffic, where the traffic is much less polluted by the Sasser worm. This example illustrates the benefit of the adaptive method since selecting a fixed value for the time interval of the basic method is challenging.

6.1.2 Accuracy gain

The only parameter of the adaptive method is the picture size, and by setting it to three different values, namely 1024, 2048, and 4096, the same high accuracy score is observed, 0.99, 0.98, and 0.99, respectively. However, the number of reported alarms decreases as the picture size increases, which is 373, 173, and 117 events respectively. Thus, for the following experiments the picture size is set to 1024 in order to report as much anomalous traffic as possible.

The comparison between the two versions of the method emphasizes the better false positive and true positive rates of the adaptive method. Namely, it identifies 369 source addresses infected by Sasser (i.e. 86% of the Sasser traffic manually identified). However, the basic method, with identical parameters but a fixed time interval of 10 seconds, identifies only 258 source addresses related to Sasser (i.e. 60% of the Sasser traffic manually identified). The basic version of the method is able to identify the same amount of Sasser traffic only if the time interval is set to one second, however, in this case 229 http traffics were also reported and a manual inspection revealed that they are benign traffic regarded as false positive alarms.

7. EVALUATION

The adaptive detection method is evaluated by analyzing four years of MAWI traffic (i.e. 2003, 2004, 2005, and 2006) and comparing its results to the outputs of three other anomaly detectors based on different theoretical backgrounds.

7.1 Compared detectors

For performance comparison we select three detection methods that are, similarly to the proposed method, analyzing only packet header and aim at finding nonspecific classes of anomaly. These three compared detectors are (1) the well-known PCA-based detector [14] (in this work the implementation of this detector relies on sketches to analyze traffic taken from a single link [13]), (2) the detection method based on multi-scale gamma modeling and sketches [6], and (3) the detector based on the Kullback-Leibler (KL) divergence and association rule mining [4]. The picture size parameter of the adaptive method is set to 1024, whereas, the parameters of the three other methods are set with fixed and arbitrary values that are globally suitable for the analyzed MAWI traffic.

The four detectors commonly aim at finding any kinds of traffic anomaly by inspecting only IP header, however, they stand for separate classes of anomaly detector as they aggregate traffic using different formalisms and rely on distinct approaches. Namely, the proposed method monitor the traffic using pictures, whereas, the PCA-based one analyzes traffic matrices and the gamma and KL detectors take advantage of histograms. Although the gamma and KL detectors are both representing traffic in histograms, they are fundamentally different; the gamma-based detector is looking for outlier traffic whereas the KL-based one is a predictive method reporting abnormal traffic variances. Therefore, comparing the proposed adaptive method to these three anomaly detectors permits a reliable evaluation.

7.2 Reported anomalies

This section inspect the anomalies that are reported by the proposed adaptive detection method in order to evaluate its true and false positive ratio. Due to the lack of ground truth data (i.e., backbone traffic with annotated anomalies) the performance of the proposed method is evaluated using two methodologies; (1) A coarse-grained evaluation with prominent anomalies manually identified in the traffic. (2) A fine-grained comparison of the accuracy using three other detection methods and inspection of the traffic reported by
Figure 8: Application breakdown of the analyzed traffic and the results of the proposed method.
the detectors and labeled as attack by the heuristics of Table 2.

Prominent anomalies.

We manually inspected several characteristics of the analyzed traffic to identify the prominent anomalies that have to be reported by the detection method. Figure 8 displays two characteristics of the analyzed traffic, namely the percentage per application of transmitted packets and bytes. The application corresponding to each traffic is recovered using the CoralReef port-based classifier [1].

We identified five main events that have significantly affected the characteristics of the MAWI traffic from 2003 to 2006 (Fig. 8a and Fig. 8b). Four events are identified by inspecting the percentage of transmitted packets per application; from August 2003 to January 2004 we observed a substantial number of ICMP flows constituting a long-lasting ping flood. The spreading of the Blaster worm is also observed from August 2003 in the MAWI traffic. Another worm called Sasser is observed from June 2004 to June 2005 in the form of three peaks representing three outbreaks of different variants of the worm. After the update of the link in July 2007, an important traffic against DNS servers is observed. This traffic is particularly intense in the middle of November 2006 (e.g., the DNS traffic measured on the 2006/11/11 stands for 83% of all packets recorded this day). Regarding the percentage of transmitted bytes per application another event is observed from August 2003, it corresponds to the outbreak of a email-based worm, called Sobig.

The traffic transmitted by the three worms manually identified in the analyzed traffic (i.e., the Sobig, Blaster and Sasser worms) are successfully reported by the proposed adaptive method (Fig. 8c). Since these worms spread in the network by contacting a substantial number of peers, the corresponding traffic highlights an abnormal dispersion in the destination IP address space that is easily identified by the proposed method. The adaptive method also effectively identifies the DNS flood appeared at the end of 2006 (Fig. 8c). This traffic is characterized by numerous hosts initiating several connections to a few servers. Thereby, the proposed method successfully detect this anomalous traffic because of its concentration in the destination IP address space and its distribution in the source IP address space.

Although the properties of the traffic have significantly varied over the four years (particularly after the link update), the proposed adaptive method efficiently detected anomalous traffic without any parameter adjustment from network operators.

Accuracy and attacks breakdown.

Based on the heuristics of Table 2, the proposed adaptive method is evaluated by accuracy comparison with the three other detection methods.

Figure 7 shows the accuracy achieved by the four detectors for each year of analyzed traffic. The average accuracy of the proposed method is higher than the one of the three other detectors during the four years of MAWI traffic. Among the three other detection methods the KL-based one is the best detector in terms of accuracy; moreover, it occasionally outperforms the method proposed in this article (Fig. 7b and Fig. 7c).

The circumstances in which the KL-based detector remarkably outperforms the other detectors were thoroughly inspected and this highlighted the fact that this detector reports a high ratio of attacks but out of only a small number of alarms. Consequently, the KL-based detector achieves a high attack ratio along with a high false negative rate (i.e., missed anomalies). Figure 9 shows the quantity of attacks reported by each detector classified with the labels from Table 2 (RPC is omitted as only 11 alarms of this kind were identified in the four years of traffic) and emphasizes the large amount of anomalies missed by the KL-based one.

The PCA and Gamma-based detectors, however, report the same quantity of attacks as the proposed method along with numerous alarms classified as special (Fig. 7). Although the proposed method is more sensitive to Sasser and attacks towards NetBIOS services, the Gamma-based method detected slightly more unusual ping traffic (66 alarms) and traffic labeled as flood (337 alarms) for the four years of analyzed traffic. Nevertheless, the PCA and Gamma-based detectors were considerably worse than the adaptive method in terms of accuracy, and this drawback is due to the quantity of traffic classified as special that was reported by these two detectors (i.e., high false positive rate).

The advantage of the adaptive method is to consistently adapt its time interval over the four years of analyzed traffic, and therefore, it constantly detects a large quantity of anomalous traffic while the number of reported benign traffic is low.

7.3 Missed anomalies

In order to highlight the limits of the proposed method this section inspects its false negative ratio, namely the proportion of anomalies that are missed by the proposed detection method. Nevertheless, due to the lack of ground truth data identifying the missed anomalies is a challenging task. The two following methodologies help us to pinpoint anomalous traffic that is not reported by the proposed detector; (1) A coarse-grained evaluation with prominent anomalies manually identified in the traffic. (2) A fine-grained inspection of anomalous traffic reported by the three compared detection method (i.e., Gamma-based, KL-based and PCA-based) but not by the proposed adaptive method.
Prominent anomalies.

The manual inspection of the analyzed traffic revealed five prominent anomalies of which one is partially missed by the proposed adaptive method, that is the ping flood emerged in 2003 (Fig. 8a and 8c). This significant ping flood is characterized by numerous point to point high-rate flows (hereafter referred as alpha flows) using the ICMP protocol that are difficulty detectable by the proposed method for several reasons. First, since ICMP traffic have no port information it is only monitored in one of the five picture categories. Second, this traffic mainly consists of a set of long-lasting point to point flows without common source or destination, thus, preventing it to be shown as a line in analyzed pictures. Finally, the typical characteristic highlighting this anomalous traffic is the substantial number of transmitted packets whereas this feature is not monitored by the proposed detection method.

Attacks detected by other detectors.

We investigate the results of the three compared detection methods (i.e., Gamma-based, KL-based and PCA-based) to uncover the false negative rate and shortcomings of the proposed adaptive detection method. Since there is a low probability for a benign traffic to be reported as anomalous by the three compared detection methods, we consider a traffic as false negative if it is reported by all the detection methods but not by the proposed one and it is categorized as attack by the heuristics of Table 2.

As shown in Figure 10, 80% of the anomalous traffic missed by the proposed detection method is labeled as ping or flood by the heuristics. This traffic is mainly composed of alpha flows containing numerous Ping or SYN packets and representing one-to-one connections (contrarily to the successfully reported traffic from worms or DDoS attacks standing for one-to-many or many-to-one connections). These one-to-one connections appear in the analyzed pictures as single points and are difficult to identify using the proposed detection method. Furthermore, another characteristic of these flows is that they account for a large fraction of the total number of packets or bytes, however, these two traffic features are not monitored by the proposed detection method. Since the proposed detection method is focusing on the distribution of the traffic features but not the volume of the traffic this method is insensitive to alpha flows. Also the proposed adaptive parameter tuning cannot overcome this shortcoming as it is inherent to the theoretical background of the proposed detection method. The class of anomaly misdetected by the proposed detector is however easily identifiable with a rate-based detection method that is monitoring the traffic volume. Therefore combining the proposed detection method and a rate-based detector would permit to detect a wider range of anomalies.

8. DISCUSSION

In addition to propose an adaptive detector, this article reveals general considerations that have to be taken into account in the domain of network traffic anomaly detection. The results presented in this article emphasizes the need of maintaining anomaly detectors parameter set optimally tuned. Indeed, Section 5.3 demonstrates that the performance of the anomaly detection method using fixed parameters is deteriorated when the characteristics of the traffic fluctuates (e.g., variations of traffic volume). Moreover, since anomalous traffic significantly alters the characteristics of the traffic anomaly detectors underperform especially during substantial anomaly outbreak. Consequently, adjusting the parameter set in regard to the fluctuations of the traffic is required to maintain the effectiveness of the detection method. These adjustments are enabled by investigating the relations between the theory underlying the detection method and the characteristics of network traffic.

The evaluation of the proposed adaptive detection method validates the efficiency of the adaptive mechanism to optimally set the parameters of the detector. Although this adaptive mechanism ensures the anomaly detector to perform optimally we observed that a certain class of anomaly is still misdetected by the proposed detector. This shortcoming is inherent to the design of the detection method thus independent from its parameter set tuning. In general, each anomaly detection method is expected to have weaknesses in detecting certain classes of anomaly; however maintaining its parameter set optimally tuned ensures that the detector is efficiently detecting the classes of anomaly it is designed for.

Section 7.3 highlights the shortcomings of the proposed detector and describes the class of anomaly undetectable by this anomaly detection method. This identification of the detection method shortcomings is a crucial task that allows us to understand the limits of this detector and ease the selection of a complementary detection method that would overcome the identified shortcomings. Consequently our results support the benefits of combining anomaly detectors [17, 8, 2].

9. CONCLUSIONS

This article proposed a new anomaly detection method that takes advantage of image processing techniques to identify the flows with abnormal traffic feature distributions. Crucial challenges rarely addressed in the appropriate literature were uncovered by investigating the major drawbacks of this method; the sensitivity of anomaly detectors to traffic variations and the role of the time scale in anomaly detec-
tion. Addressing these two issues resulted in a significant improvement for the proposed detection method that overcomes any adverse conditions as it analyzes traffic within a time interval that is automatically adapted to the traffic throughput and the distribution of traffic features.

The evaluation of this adaptive method using real Internet traffic highlighted its ability to maintain a high detection rate while the traffic was significantly altered by anomalies. Therefore, these experiments indicated that the adaptive time interval enabled 26% more worm traffic to be detected, and decreased the false positive rate. The adaptive detection method proposed in this paper is also validated by comparing it with three other detection methods and using four years of real backbone traffic. The results highlighted that the proposed adaptive method allows for the detection of almost all the anomalies reported by the other detectors while it achieves the lowest false positive rate. We identified a class of anomaly that is disregarded by the proposed detection method and discussed the benefit of complementary detection methods to overcome these shortcomings, however, the study of combining several anomaly detection methods is left for future works.

Acknowledgments
The authors are thankful to Kenjiro Cho, Yosuke Himura, Yoshiki Kanda, Patrice Abry, Pierre Borgnat, and Guillaume Dewaele for their valuable comments.

10. REFERENCES

About the authors:

Romain Fontugne received his master's degree in Computer Science from Joseph Fourier University, France, in 2008. He is now a Ph.D. candidate at the Graduate University for Advanced Studies (SOKENDAI) in Tokyo.

His research interests are Internet traffic analysis and Internet security.

Kensuke Fukuda is an associate professor at the National Institute of Informatics (NII). He received his Ph.D degree in computer science from Keio University at 1999. He worked in NTT laboratories from 1999 to 2005, and joined NII in 2006. In 2002, he was a visiting scholar at Boston University. Concurrently, he is a researcher of PRESTO JST (Sakigake) since 2008.

His current research interests are Internet traffic measurement and analysis, intelligent network control architectures, and the scientific aspects of networks.
Preserving Privacy in Wireless Sensor Networks using Reliable Data Aggregation

Farzana Rahman  
MSCS Dept.  
Marquette University  
Milwaukee, WI, USA.  
frahman@mscs.mu.edu

Md. Endadul Hoque  
MSCS Dept.  
Marquette University  
Milwaukee, WI, USA.  
mhoque@mscs.mu.edu

Sheikh Iqbal Ahamed  
MSCS Dept.  
Marquette University  
Milwaukee, WI, USA.  
iq@mscs.mu.edu

ABSTRACT
An important topic addressed by the wireless sensor networks community over the last several years is the in-network data aggregation. It is significant as well as a challenging issue to provide reliable data aggregation scheme while preserving data privacy. However, in WSNs, achieving ideal data accuracy is complicated due to collision, heavy network traffic, processing delays and/or several attacks. The problem of gathering accurate integrated data will be further intensified if the environment is adverse. Hence how to attain data privacy and perfect data accuracy are two major challenges for data aggregation in wireless sensor networks. To address this problem, we propose in this paper a new privacy preserving data aggregation scheme. We present REBIVE (REliable prIVate data aggregEation scheme). In REBIVE the data accuracy maintenance and data privacy protection mechanisms work cooperatively. Different from past research, our proposed solution have the following features: providing privacy preservation technique for individual sensor data and aggregated sensor data; maintaining perfect data accuracy for realistic environments; being highly efficient; and being robust to popular attacks launched in WSNs.

Categories and Subject Descriptors
K.4.1. [Computers and Society]: Privacy.

General Terms
Measurement, Design, Reliability.

Keywords
Sensors; Data aggregation; Privacy; Reliable; Secret key.

1. INTRODUCTION
The emergence of Wireless Sensor Networks (WSNs) with special capabilities such as the ability to control the activation of different hardware and the developments in low-power computational components will bring lots of envisioned diverse applications into reality [1]. WSNs involve plenty of small and low-cost sensor nodes with limited abilities of data sensing, wireless communication, on-board processing and power supply. The application of wireless sensor networks may range from military (e.g., surveillance, intelligence gathering) to environmental (e.g., habitat monitoring, wild-life tracking). They can be deployed in shopping malls, office buildings, industries, hospitals or homes to monitor the working status of machineries, the air, light, sound, humidity, or temperature conditions of rooms, the water and electricity consumptions of homes, or the blood pressures of human beings. Though these applications can promise attractive features to the end users, they can also raise privacy concerns, for example private information leakage.

Privacy preservation can be accomplished using different cryptographic techniques. However, they cannot be applied in sensor networks because the cryptographic mechanisms can involve heavy computation which is infeasible for resource constrained sensors. Therefore, privacy can be preserved by employing different data aggregation techniques in sensor nodes. In certain contexts, when sensor readings might reveal inappropriate information about a location or an individual, data aggregation may play an essential role for alleviating privacy concerns. Data aggregation [2] is an efficient mechanism in query processing in which data are processed and aggregated within the network. Traditionally, data sampled at the sensor nodes needs to be conveyed to a base station for further processing, analysis, and visualization by the network users. Aggregation in this context can refer to the computation of statistical means and moments, as well as other cumulative quantities that summarize the data obtained by the network. Here, typical aggregation functions are used including SUM, AVERAGE, MAX/MIN, Median, Histogram, and so on [3]. Uses of such functions are important for data analysis and for obtaining a deeper understanding of the data.

Uses of data aggregation techniques in WSNs drastically reduce data communication, which consequently decrease bandwidth consumption and energy loss. Security is a key requirement for the sensor nodes to enable in-network data aggregation. Secure communication requires sensor nodes to encrypt any sensed data prior to its transmission. In addition, it is desirable to have end-to-end security with the data decrypted only at base station in order to avoid security problems as much as possible. However, data aggregation protocols demand intermediate nodes to process packets to identify the redundant ones which require data packets to be decrypted. This can be done by encrypting data with keys known by forwarding nodes. However, this will allow the forwarding nodes to retrieve private information. Therefore, these
two conflicting goals require data aggregation algorithms to be designed together with secure communication algorithms so that they can preserve privacy.

The issue of providing privacy preservation and data aggregation are two conflicting goals in WSNs which did not achieve much consideration from researchers until He et al. proposed two privacy preservation schemes in [4]. The authors proposed two schemes, cluster-based private data aggregation (CPDA) scheme and the slice-mix aggregate (SMART) scheme, for additive data aggregation in wireless sensor networks. CPDA leverages clustering schemes and algebraic properties of polynomials. It has the advantage to enable neighbor monitoring within a cluster. The second scheme, SMART builds on slicing techniques and the associative property of addition. It has the advantage of incurring less computation overhead for privacy-preserving data aggregation. The pioneering research on data aggregation continued with the proposal of collusion-resilient approximate data aggregation technique [5] by Zhang et al. presents to protect data privacy in WSNs. This technique supports data aggregation in queries targeted at special sensor data.

Both of these techniques [4], [5] have their own limitations. For example, they will not provide 100% accurate aggregation results if there is data loss in the network. Both these literatures did not consider a powerful attack model where an adversary can try to make the entire system malfunction. In such situations both the techniques will perform poorly and will result in inaccurate aggregated data. However, in some real adverse environment, for example - military environments, surveillance systems, achieving high data accuracy is very important though there is a chance of high data loss. With the advent of time, research on data aggregation technique received more attention and it was further advanced when He et al. proposed iCPDA [6], which piggybacks on a cluster-based privacy-preserving data aggregation protocol (CPDA). The same authors proposed iPDA [7], which addresses both privacy and integrity of data aggregation by constructing disjoint aggregation trees. The only added new feature in these two protocols is the preservation of data integrity. Therefore iPDA and iCPDA are not free of the drawbacks that exist in PDA [4] as they are designed on top of PDA.

To address the issue of inaccurate aggregated data because of data loss, in this paper we propose REBIVE (REliAble prIVate data aggrEgAtion) that can protect privacy at the same time ensure perfect data accuracy. It is a cluster based approach where nodes in the network form a cluster to perform the data aggregation. Nodes share their private data using a secret sharing threshold scheme. A node divides its private data into n shares such that this data can be retrieved from k out of these n shares, where n is the cluster size. The node shares these n shares with the other cluster members including the cluster head. In this way, every node within the cluster receives a share from all other nodes within the cluster. Each node aggregates all the shares received from others and replies back to cluster head. Now cluster head retrieves the aggregated version of all the private data of the nodes within the cluster and finally sends back to the query server.

The rest of the paper as follows. In section 2, we discuss the motivation behind data aggregation in sensor network. Section 3 describes the models and background. Section 4 elaborates our proposed scheme. We evaluate the performance of REBIVE in section 5. Related work is summarized in section 7. Finally, section 7 concludes the paper.

2. MOTIVATION AND DESCRIPTION OF DATA AGGREGATION IN WSN

Given that sensor nodes are severely constrained in terms of battery lifetime, the aim is to always come up with techniques to increase such lifetime. Battery power is needed by a sensor node to perform computations with its CPU and, most importantly, to transmit and receive packets through its radio. Thus, an obvious solution would be to reduce the CPU usage and the number of transmission and receptions. Energy consumed for communication (i.e., transmission and reception) purpose is much higher than computation. Therefore, it is wise to focus on reducing the communication costs and promote more computation than communication. Such a reduction in the number of transmissions and receptions will also increase the available bandwidth, which is also a scarce source for WSNs. There are many opportunities in WSNs to reduce the number of transmission and receptions because WSNs are characterized by huge numbers of nodes with considerable redundant data because of the similarity of reports sensed by sensor nodes at nearby locations. Other cases in which the number of communications can be reduced are aggregate queries. These are queries where the result is computed based on the data received from every node. For example, if a query is asking for the average temperature in a region, then each sensor’s value should be received at the base station and averaged. However, there is no need for the base station to receive all of the values. Instead, the average can be calculated within the network while the packets are traveling through the base station. A node can get the values from its neighbors, compute the average (i.e., create one packet out of n), and send the average to the upper nodes.

There are several applications in which data aggregation can be employed. But mostly, in sensor databases, data aggregation is a crucial technique for performing aggregate queries. However, as sensor network applications expand to include increasingly sensitive measurements of everyday life, preserving data privacy becomes an increasingly important concern. For example, a future application may measure private household information such as energy usage patterns. Without providing proper privacy protection, such applications of WSNs will not be practical, since participating parties may not allow tracking their private data. In this paper, we discuss how to carry privacy-preserving data aggregation in wireless sensor networks reliably.

3. MODELS & BACKGROUND

3.1 Design Objectives

Accomplishing data aggregation in WSNs with reliability and privatesly is a major concern. We summarize the fundamental characteristics that a privacy preserving data aggregation scheme (DAS) has to comply with.

**Data privacy:** Every sensor’s individual data (raw data) should be known only to itself. Even the intermediate aggregated data should be kept private from other sensors in the network. Disclosure of any such data can leak private information (e.g. users’ household activities). In order to preserving privacy, the links between sensors must be kept secret to prevent the outsiders from eavesdropping. Since the insiders (in-network sensors) can decrypt the data, the private aggregation scheme should be robust to uphold privacy of the whole network.
**Competition:** DAS has to be efficient in terms of resource and power consumption. Since communication is more costly than computation in WSNs, in-network aggregation can reduce bandwidth requirement. Introducing privacy preservation in data aggregation schemes imposes some overhead on sensors in terms of communication and computation. A good privacy preserving DAS has to keep the overhead to a minimum.

**Data accuracy:** Data accuracy means how accurate the queried result is. It is evidently desirable that data aggregation schemes should achieve nearly ideal accuracy (i.e., 100% accuracy), but of course, not at the cost of private information (raw and/or intermediate aggregated data). Since data accuracy is a performance metric, the higher data accuracy of a private DAS increases the acceptance of the scheme in real world.

Robust against attacks: WSNs can undergo several malicious attacks. An attacker goal is two folded here. On one hand, the attacker wants to discover and disclose individual data of any sensor and thus violate privacy. Very common and simplest attack is eavesdropping over wireless links. He can even compromise some sensors that will collude to reveal the private data of other sensor(s). On the other hand, the attacker can intrude the sensor network to make the queried result inaccurate by interrupting the aggregation process so that the system will malfunction. This situation can occur when some compromised sensors reply with false data or remain silent for small time spans during the aggregation process. By keeping the wireless links secret, eavesdropping can be prevented. A private data aggregation scheme should prevent such attacks.

### 3.2 Network Model

We consider a sensor network consisting of $N$ sensors. In this paper, we assume the network as a connected graph $G = (V,E)$, where $V$ is the set of sensors and $E$ implies the wireless links between any two sensors. From now on, we use sensor and node interchangeably to refer to a sensor of the network.

A data aggregation function is defined over all the sensors individual data such as $f(t) = \mathcal{A}(v_1(t), \ldots, v_N(t))$, where $v_i(t)$ is the individual data of $i$th sensor at time $t$. In this paper, we assume $v_i \in \mathbb{F}_p, \forall i \in [1,N]$, where $P$ is an large prime integer and $P > N$. Query server (sometimes called data sink) issues the query that spread out over the whole network. $f(t)$ is calculated at the query server after it receives all the replies from sensors in response of its query placed at time $t$. $\mathcal{A}()$ can be any aggregate function such as $\text{Sum, Count, Mean, Median, Max/Min, Variance, Standard Deviation, Histogram}$, etc. In this paper, we consider the additive aggregate function such as $\mathcal{A}(t) = \Sigma_{i=1}^N v_i(t)$. The logic behind this selection is that many other aggregate functions can be reduced to summation.

### 3.3 Assumptions

We assume the raw data measured by a sensor is an integer from the finite field $\mathbb{F}_p$, where $P$ is a large prime. To prevent attackers from eavesdropping, we assume some messages are transmitted after encrypting. In our model we use a random key distribution mechanism proposed in [9]. According to [9], any pair of nodes can share at least one common key with high probability. If there is no direct link between a pair of node, they can be assigned a path key.

### 3.4 Attack Model

In the system model we mentioned earlier, we consider an attack model. Evidently we should not consider an attacker as weak as passive attackers. However, considering an attacker much powerful so that it can launch any type of attack is not realistic. A practical attack model is appropriate to analyze any private data aggregation scheme. Our attacker can launch passive attacks like eavesdropping over wireless links as well as some active attacks such as node collision. The colluded nodes help the attacker to discover raw data of other sensor(s), since he can sniff all the data coming to and/or coming out the colluded nodes. Even the attacker can launch an attack that causes the compromised nodes to remain silent for small time span(s) while the aggregation is going on. This malicious attack is launched with the intention of generating inaccurate aggregated data (in our case, inaccurate $\text{Sum}$) so that the whole system starts malfunctioning and becomes unreliable.

### 4. RELIABLE PRIVATE DATA AGGREGATION SCHEME

Now we present our data aggregation protocol. The protocol is private in a sense that it can preserve data privacy of individual sensor and of the whole network. On other hand, it is reliable since it can provide perfect data accuracy even when there is some data loss in the network. Ours is cluster-based approach that undergoes the steps — query launch & clusters formation, data aggregation at sensor nodes, and post-aggregation at query server.

#### 4.1 Query Launch & Clusters Formation

The query server triggers the formation of clusters by launching a query. Thereafter all the sensor nodes collaborate to form clusters within the network. Here we briefly review the clusters formation technique described in [4] on which REBIVE operates. The server launches query with a HELLO message. Upon the reception of the HELLO message, each sensor decides to be a cluster head depending on a probability $p_c$, which is a predefined system parameter. If a node becomes a cluster head, it forwards the HELLO message to its neighbors. Otherwise, it just waits for a predefined time span and finally joins one of the clusters by sending a JOIN message. Thus multiple clusters are formed in the network. To maintain data privacy, cluster size has to be $n \geq 3$. This is a limiting factor of REBIVE.

#### 4.2 Data Aggregation within Clusters

Now the nodes within a cluster collaborate with each other to perform data aggregation. Since a cluster is a group of mutually suspicious individuals (sensors) and the raw data of any individual (sensor) is private information of the owner, we prefer a threshold scheme to share sensor private data. Before moving on to the description of the calculation for data aggregation within a cluster, we want to focus on the background of a threshold scheme that REBIVE adapts for private data sharing.

#### 4.2.1 $(k,n)$ threshold scheme

Here we present a brief overview of a $(k,n)$ threshold scheme proposed in [10]. This scheme is also known as Shamir’s Secret Sharing. In this scheme, a secret data $S$ is divided into $n$ shares $S_1, \ldots, S_n$ in such way that:

a) Any subset of $k$ shares $S_1, \ldots, S_k$, where $k \leq n$, can reconstruct the secret data $S$. 

---

Applied Computing Review 54
b) But any subset of \( k - 1 \) shares \( S_{i_1}, \ldots, S_{i_{k-1}} \) reveals absolutely no information about \( S \).

If \( k = n \), then all shares are needed to reconstruct \( S \).

This scheme is based on polynomial interpolation. Suppose the secret data \( S \) is in a finite field \( \mathbb{F} \) and \((k - 1)\) random coefficients \( a_1, \ldots, a_{k-1} \) from \( \mathbb{F} \). Therefore a polynomial of degree \((k - 1)\) can be \( y = f(x) = a_0 + a_1x + \cdots + a_{k-1}x^{k-1} \), where \( a_0 = S \). Now the party can compute \( n \) shares for distinct \( x_1, \ldots, x_n \), where \( \forall i, x_i \neq 0 \), and distribute the shares among the participants.

In the reconstruction, if any subset of \( k \) shares are given such as \((x_i, y_{ij})\), where \( j \in \{1, \ldots, k\} \) and \( i \in \{1, \ldots, n\} \), then the coefficients of the polynomial can be evaluated by interpolation, and thus \( S = a_0 \) can be retrieved. However any subset of \((k - 1)\) shares cannot reconstruct the polynomial and therefore leaves \( S \) completely undetermined.

4.2.2 REBIVE adapts secret sharing

Since there is a subtle difference between providing secrecy and preserving privacy of a data, REBIVE adapts the \((k, n)\) threshold scheme for protecting data privacy. In case of secrecy, only the authorized party/parties can know (or retrieve) the secret data; but in case of privacy, a private data must be known only to its owner.

In our scheme, data can be aggregated within clusters by 4 phases – broadcasting \( x_i \)'s, sharing \( y_i \)'s, reporting \( \sum y_i^* \), and recovering \( \sum S \). By describing these phases, we illustrate how \((k, n)\) threshold scheme can be used for data privacy in case of additive aggregation function. For the ease of description, we consider a cluster of \( 4 \) nodes – \( A, B, C \) and \( D \), as shown in Fig. 1. \( S_A, S_B, S_C, S_D \) denote the raw data of the sensors, respectively, where each \( S_i \in \mathbb{F}_p \). Our objective is to relay the sum of these raw data that is \((S_A + S_B + S_C + S_D)\), to the cluster head, say, node \( A \), without passing on individually any of the raw data.

a) Broadcasting \( x_i \)'s: The cluster head, in our example, node \( A \) broadcasts a message such as \( \langle x_1, \ldots, x_n, x_i, k \rangle \), where \( n \) is the cluster size, \( k \) denotes the total number of shares needed for reconstruction, and \( 3 \leq k \leq n \). The set \( \{x_j| 1 \leq j \leq n\} \) specifies the values at which the shares will be calculated. By \( x_i, A \) tells the other nodes within the cluster that he wants to accumulate \( y_i \)'s values calculated for \( x_i \). In our example, \( A \) broadcasts \( \langle x_1, x_2, x_3, x_4, x_5, 3 \rangle \). Now each cluster member excluding the cluster head has to announce an \( x \)'s value from the set of \( \{x_1, \ldots, x_n\} \) so that each one gets an unique \( x \). Hence \( B \) announces an \( x \)'s value for himself by the message \( \langle x_i \rangle \), where \( x_i \in \{x_1, \ldots, x_n\} \setminus \{x_1, x_i\} \). In Fig.1(a) \( B \) announces \( \langle x_1 \rangle \). Node \( D \) selects an \( x \)'s value from \( \{x_1, \ldots, x_n\} \setminus \{x_1, x_i\} \). For say, it is \( x_3 \) in our example. So \( C \) will be accumulating for the remaining one, \( x_4 \).

b) Sharing \( y_i \)'s: Node \( A \) forms a polynomial of degree \( k - 1 \). In our example the polynomial is of degree 2 like \( y^A = f_2(x) = a_0 + a_1x + a_2x^2 \), where \( a_1, a_2 \) are randomly chosen from a uniform distribution over \( \mathbb{F}_p \) by node \( A \) itself and \( a_0 = S_A \). These coefficients are known only to node \( A \). Now \( A \) computes \( y_i^A = f_2(x_i), \forall i \in \{1, 2, 3, 4\} \). Similarly, \( B \) randomly selects \( b_1, b_2 \in \mathbb{F}_p \) and constructs his polynomial such as \( y_i^B = f_3(x_i) = b_0 + b_1x + b_2x^2 \), where \( b_0 = S_B \); finally evaluates \( y_i^B = f_B(x_i), \forall i \in \{1, 2, 3, 4\} \). In the same way, node \( C \) and \( D \) calculates \( y_i^C \) and \( y_i^D \) for all \( i \in \{1, 4\} \). Now node \( A \) sends encrypted \( E(y_{i}, k_{AB}) \) to \( B \), \( E(y_{i}^C, k_{AC}) \) to \( D \) and \( E(y_{i}^D, k_{AD}) \) to \( C \). For the ease of illustration, Fig.1(b) depicts messages in and out of only node \( A \). As the sharing \( y_i \)'s values go on, \( A \) receives \( \{y_2^A, y_3^A, y_4^A\} \), \( B \) obtains \( \{y_1^B, y_2^B, y_3^B\} \), \( C \) and \( D \) gets \( \{y_1^C, y_2^C, y_3^C\} \) and \( \{y_2^D, y_3^D, y_4^D\} \), respectively.

![Figure 1. Illustration of data aggregation of REBIVE](image-url)
accumulated y’s. Therefore $B, C, D$ calculates and broadcasts $y_1, y_2, y_3$, respectively (see Fig.1(c)).

d) Recovering $\Sigma S$: Being the cluster head, it’s the responsibility of node $A$ to recover the summation of the raw data of the sensors within the cluster. $A$ populates the set of shares as $\{(x_i, y_i)\} | i \in [1,4]$. By utilizing the reconstruction phase of $(k,n)$ threshold scheme and using any subset of the set of shares such as $\{(x_i, y_i)\} | i \in [1,3] \text{ and } j \in [1,4]$, node $A$ can reconstruct a polynomial $q(x)$ of degree $(k-1)$ (for our example, we use Lagrange basis polynomial and the generated polynomial is a quadratic function) and thus evaluates intermediate aggregated value $\Sigma S = S_A + S_B + S_C + S_D = q(0)$. Therefore $A$ deals with $\Sigma S$, but never comes across any individual $S$, as long as it is guaranteed that $3 \leq k \leq n$. Hence data privacy is preserved within a cluster.

4.3 Post-Aggregation at Query Server

Now it’s time for the cluster heads to pass on their intermediate aggregated values to the query server. The cluster heads form an aggregation tree rooted at the query server. When the server receives all these intermediary sums, he just adds up the values. Thus the query server is reported with the sum of raw data of sensors while preserving each individual sensor’s privacy.
In this section we evaluate our proposed scheme. Our assessment is done in terms of privacy preservation of any node in the network.

Private data of a sensor node $q$ can be disclosed to an attacker in two ways. One way, the attacker compromises multiple neighboring nodes of $q$ so that they jointly disclose the private data of $q$. On the other way, the attacker learns $k$ out of $n$ shares of the private data by eavesdropping and reconstructs the private data from the $k$ shares. For this paper, we consider that the attacker will apply the latter way. To learn any $k$ shares out of $n$, the attacker needs to decrypt the $k$ outgoing messages from the sensor node $q$. Therefore the privacy of a sensor node becomes $\psi(y)$ that is the probability of successfully cracking the private data of any node (say, $q$) where $y$ is the probability of decrypting the link level message. To decrypt a link level message, the attacker has to know the secret key that $q$ shares with the other node of the link. So in other words, $y$ is the probability that the secret key is known to the attacker.

Figure 2. Measurement of privacy disclosure in a dense network
Given $n$ is a cluster size and node $q$ is a member of the cluster, according to REBIVE $q$ shares its private data with $(n-1)$ other nodes within the cluster. But any $k$ of the shares can reveal the private data. Therefore we can define $\psi(y)$ as

$$\psi(y) = \sum_{t=0}^{C_{\text{max}}} P(n = t) (1 - (1 - \delta)^{n}) - \sum_{k=C_{\text{min}}}^{t} \binom{t}{k} y^{k}(1 - y)^{t-k}$$

where $C_{\text{min}}, C_{\text{max}}$ are the minimum and maximum cluster size, respectively. $P(n = t)$ represents the probability that the cluster size is $t$.

Fig. 2 shows the privacy preservation performance of REBIVE in different scenarios via simulation in MATLAB. We consider a network of 500 nodes located in 400meters x 400meters area. We generate two types of network: dense and sparse. Each point means the average of 50 runs of the simulation. With the decrease in $p_c$, less clusters form with larger size. Therefore variation of any $k$ out of $n$ shares increases with the increase in cluster size. So the chances of the private data being disclosed will be high. Fig.2(a) shows the privacy disclosure probability of a node in dense network. We can see from these graphs (fig 2(a), 2(c), 2(e), 2(g), 2(b), 2(d), 2(f), 2(h)) that as we increase the minimum cluster size, the probability of privacy disclosure decreases.
In a sparse network $\psi(y)$ (see Fig.3(a), 3(b), 3(c), 3(d), 3(e), 3(f), 3(g), 3(h)) is much lesser than that in a dense network while keeping other parameters same. By varying $k = \lceil n/2 \rceil$ to $C_{\text{max}}$ we get much lower $\psi(y)$ (see Fig.2(c)) because we are not considering $k = C_{\text{min}}$ to $\lceil n/2 \rceil -1$ that favors the attacker. We see that with the increase of minimum cluster size, in sparse network, the probability of privacy disclosure decreases.

6. RELATED WORKS
Data aggregation in WSNs can play an important role to save resource of tiny sensors as they have very limited resource and battery power. So, the key purpose of data aggregation is to achieve bandwidth and energy efficiency, which have been proved to be highly beneficial for resource limited sensors [3]. It is significant as well as a challenging issue to provide reliable data aggregation scheme while preserving data privacy. Lots of works have been done on data aggregation in WSNs including [11-17].
All these techniques assume that the underlying communication medium is secure and all sensors are trusted. However, in reality, sensors are likely to be deployed in untrusted environments where an adversary may launch several attacks simply by eavesdropping or compromising a sensor.

Privacy preservation has also been an important issue in the domain of data mining for long and therefore much works [18–20] have been done in this area. Two major classes of schemes are used for this purpose. The first class is based on data perturbation techniques. However, data perturbation techniques have the drawback that they do not yield accurate aggregation results and so they might not preserve privacy well [19]. SMC is another technique that is used to preserve privacy [21, 22]. It performs a joint computation of a function with multi-party private inputs. However, SMC [23, 24] cannot be applied to resource-constrained WSNs as they are computationally expensive.

Some previous works [11–16] address data aggregation techniques in various applications. Early work on in-network aggregation, e.g., TinyDB [3], uses a tree-like structure to disseminate queries and collect results back to the sink. Various algorithmic techniques have been proposed to allow efficient data aggregation [25, 26]. Some aggregation techniques try to be secure by addressing the issue of attacks where adversary tries to tamper with sensors. Przydatek et al [2] present protocols that can be used by a trusted remote user to query a sensor network in which the base station may be compromised and the base station is the only aggregator. Yang et al. [27] proposed SDAP, a secure hop-by-hop data aggregation protocol which can tolerate more than one compromised node. It uses “divide-and-conquer” and “commit-and-attest” principles. The principle “divide-and-conquer” means SDAP dynamically partitions the topology tree into multiple logical groups (subtrees) of similar sizes. In [28], Feng et al. propose a family of secret perturbation-based schemes. It can protect sensor data confidentiality without disrupting additive data aggregation result.

He et al. proposed two pioneering privacy-preserving data aggregation protocols, CPDA and SMART, in [4] which do not assume data manipulation attacks. Both techniques depend on the collaboration between neighboring nodes to hide the actual data. Both CPDA and SMART schemes have their limitations. They can only tolerate the collusion of up to a certain threshold number of sensor nodes. If more than the threshold numbers of sensor nodes collude then the technique will result in higher communication overhead. Zhang et al. presents collusion-resilient approximate data aggregation technique [5] to protect data privacy in WSNs. This technique supports data aggregation in queries targeted at special sensor data or the distribution of sensor data and queries targeted at particular sensor nodes. Both this techniques [4], [5] will not provide 100% accurate aggregation results if there is data loss in the network. But out proposed technique, REBIVE, achieves perfect data accuracy to some extent and tries to maintain 100% accuracy most of the time. Both these literatures did not consider a powerful attack model where an adversary can try to make the entire system malfunction. However, REBIVE is robust against such powerful attacks that we mentioned in our attack model. In some real adverse environment, for example - military environments, surveillance systems, vehicular networks, achieving high data accuracy is very important though there is a chance of high data loss. In such environments REBIVE achieves perfect data accuracy though [4], [5] performs poorly to achieve high accuracy.

In wireless sensor networks and recently emerged participatory sensing applications [29], [30], [31], both privacy of individual sensing data and integrity of the final aggregation trees. This allows establishing existential proof of whether or not there is data pollution for individual attackers. However, this technique cannot find out which node is malicious. But iCPDA lets the peers to monitor for which node is malicious. Both these (iCPDA and iPDA) protocols use the same technique as PDA [4] to preserve privacy. The only added new feature in these two protocols is the preservation of data integrity. So iPDA and iCPDA are not free of the drawbacks that exist in PDA. However, our protocol aims to achieve perfect data accuracy rather than integrity because data aggregation techniques can be more reliable if they have high data accuracy.

7. CONCLUSION
Privacy preserving data aggregation in wireless sensor networks is a challenging problem. In this paper, we present a data aggregation scheme for WSNs that is reliable as well as privacy preserving. In the proposed scheme a cluster head can retrieve the aggregated data of the nodes within the cluster even if data loss happens during the aggregation process. We assess the privacy preserving performance for different types and scenario of the network.

For our future work, we would like to perform assessment of data accuracy of REBIVE. Other future research direction could be designing privacy preserving data aggregation schemes for some other aggregation function.

8. REFERENCES
[6] He, Wenbo and Liu, Xue and Nguyen, Hoang and Nahrstedt, Klara, A Cluster-Based Protocol to Enforce Integrity and Preserve Privacy in Data Aggregation.
About the authors:

Farzana Rahman is a member of Ubicomp lab, Marquette University, Milwaukee, WI. Currently, she is pursuing PhD in Computational Sciences at Marquette University. She received her B.Sc in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 2008. She also received Masters in Computer Science from the Department of Mathematics, Statistics, and Computer Science at Marquette University in 2010. She was one of finalists of Google Anita Borg Scholarship 2010.

Her field of interest encompasses Security and Privacy of RFID Systems and Wireless Sensor Networks, Green Computing, Rural Healthcare Privacy and Trust and Privacy in Pervasive Computing. She can be contacted at http://www.mscs.mu.edu/~frahman

Endadul Hoque is currently pursuing his PhD in Computer Science in the Department of Computer Science at Purdue University. He received his B.Sc degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 2008. He also received Masters in Computer Science from the Department of Math., Stat., and Computer Science at Marquette University in 2010. His research interests include Security and Privacy of Systems and Networks in general; in particular, Reliable Distributed Systems, Security in Wireless Networks, Security and Privacy in RFID Systems and WSNs. He can be contacted at http://www.cs.purdue.edu/homes/mhoque/

Sheikh I. Ahamed is an associate professor in the department of Math., Stat., and Computer Science at Marquette University, USA. He is a member of the IEEE, ACM, and the IEEE Computer Society. He completed his Ph.D in Computer Science from Arizona State University, USA in 2003. His research interests are pervasive computing, security, privacy and trust issues in pervasive computing, and pervasive healthcare. He has published more than 100 peer reviewed journal, conference, and workshop papers. He has received four best papers awards in the past four years. He has served as the program chair of a number of conferences and workshops such as COMPSAC 2011, SPTSA 2006-2009. He also directs the Ubicomp research lab at Marquette University, USA. Dr. Ahamed can be contacted at sheikh.ahamed@mu.edu and http://www.mscs.mu.edu/~iq