Advances in Multimedia

Web Services in Multimedia Communication

Guest Editors: Mohamed Hamdi, Nabil Tabbane, Tai-Hoon Kim, and Sajid Hussain

Web Services in Multimedia Communication

Advances in Multimedia

Web Services in Multimedia Communication

Guest Editors: Mohamed Hamdi, Nabil Tabbane,
Tai-Hoon Kim, and Sajid Hussain

Editorial Board

Contents

Web Services in Multimedia Communication, Mohamed Hamdi, Nabil Tabbane, Tai-Hoon Kim, and Sajid Hussain
Volume 2012, Article ID 129504, 2 pages

Image Encryption Using a Lightweight Stream Encryption Algorithm, Saeed Bahrami and Majid Naderi
Volume 2012, Article ID 767364, 8 pages

A Novel k-out-of-n Oblivious Transfer Protocol from Bilinear Pairing, Jue-Sam Chou
Volume 2012, Article ID 630610, 9 pages

Parlay X Web Services for Policy and Charging Control in Multimedia Networks, Ivaylo Atanasov, Evelina Pencheva, and Dora Marinska
Volume 2012, Article ID 296234, 12 pages

Seamless Integration of RESTful Services into the Web of Data, Markus Lanthaler and Christian Gtl
Volume 2012, Article ID 586542, 14 pages

A Framework for Automatic Web Service Discovery Based on Semantics and NLP Techniques, Asma Adala, Nabil Tabbane, and Sami Tabbane
Volume 2011, Article ID 238683, 7 pages

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2012, Article ID 129504, 2 pages
doi:10.1155/2012/129504

Editorial
Web Services in Multimedia Communication

Mohamed Hamdi,[1] Nabil Tabbane,[2] Tai-Hoon Kim,[3] and Sajid Hussain[4]

[1] *Computer Science and Networks Department, School of Communication Engineering (Sup'Com), University of Carthage, Technopark El Ghazala, 2083 El Ghazala, Tunisia*
[2] *Multimedia Mobile Radio Networks Research Unit (MEDIATRON), Sup'Com, University of Carthage, Technopark El Ghazala, 2083 El Ghazala, Tunisia*
[3] *Global Vision School, 101-ho, Jeongseong-vill, 593-7, Wonseong-dong, Dongnam-gu, Cheonan-si, Chungnam-do, Republic of Korea*
[4] *Department of Business Administration, Fisk University, 321 Park Johnson Hall (PJ), Nashville, TN, USA*

Correspondence should be addressed to Mohamed Hamdi, mmh@supcom.rnu.tn

Received 29 November 2012; Accepted 29 November 2012

Multimedia communication has made a tremendous impact on web technologies and has been the basis for various applications ranging from videoconferencing to medical imaging and target surveillance. The advent of new efficient multimedia coding schemes has made it technologically feasible and economically viable to develop web services that encompass the real-time transmission of high-resolution multimedia content. For instance, Youtube users spend 3 billion hours per month on the site. Moreover, movies streamed by Netflix can reach 20 percent of US broadband traffic. This is just the beginning of a video flood that will swamp the web.

Multimedia communication and web services constitute the confluence point where the interest of service providers, telecommunication operators, and end users meet. Many digital services, such as pay-TV, confidential video conferencing, medical and military imaging systems, require reliable security in storage and transmission of digital images/videos. As the rapid progress of Internet in the digital world today, the security of digital images/videos has become increasingly important. In recent years, more and more consumer electronic services and devices, such as mobile phones and PDA (personal digital assistant), have also started to provide additional functions of saving and exchanging multimedia messages. The prevalence of multimedia technology in our society has promoted digital images and videos to play a more significant role than the traditional dull texts, which demands a serious protection of users' privacy. Because online videos are useful for a wide spectrum of web services, companies are reorganizing their software and network infrastructures to accommodate it. That focus has touched off major challenges over protocols, algorithms, and standards allowing the provision of reliable and high-quality performance in multimedia communication.

The objective of this special issue is to bring together research contributions on the design, specification, and implementation of architectures, protocols, and algorithms for current and future web services based on multimedia communication. The special issue is devoted to those areas of web technologies where real-time video and image transmission bring new insights to yield effective solutions for the problems of interest. The issue contains five high-quality papers whose topics range from image encryption to automatic service discovery offering innovative methodologies, algorithms, and theoretical results by using the existing achievements in multimedia communication and by extending them to fit the specific needs of web applications.

This special issue covers a variety of recent research articles and comprehensive materials on recent technological advances in the field of Web Services for Multimedia Communication. We invited authors to submit original research articles that held with recent advances in Web services technology with application to multimedia communication. We received 8 submissions from authors around the world. Papers that passed our preliminary screening were passed to a rigorous review process involving experts in this field. Finally, five articles were accepted for this issue.

In the paper, "*Image encryption using a lightweight stream encryption algorithm*" by S. Bahrami and M. Naderi, a simple and lightweight stream encryption algorithm for image data

is proposed. This algorithm operates on sections of the image using a different key for every section. The secret key of the encryption scheme is protected by a block cipher algorithm (such as AES). The authors discuss the robustness of the lightweight encryption scheme to meet-in-the-middle and chosen ciphertext attacks. A series of statistical tests have also been performed to give an idea on the protection level provided by the proposed encryption technique. For instance, the histogram analysis shows that the distribution of pixel brightness in the encrypted image is uniform. The speed of the encryption functions has also been shown to be convenient for real-time transmission.

The paper, "*A novel k-out-of-n oblivious transfer protocol from bilinear pairing*" by J.-S. Chou, provides an oblivious transfer protocol that can be used in various applications like the signature of fair contracts, oblivious database searches, mental poker games, privacy-preserving auctions, and secure multiparty computations. In addition to cryptographic robustness, the author addresses the complexity of the protocol, which is essential for commercial applications. The major advantage of the proposed approach is that it can run without the secure channel required by the existing techniques to enforce protection against replay, denial of service, and man-in-the-middle attacks. The correctness of the scheme has been formally proved in addition to mutual authentication, privacy, and anti-replay properties.

In the paper, "*Parlay X Web services for policy and charging control in multimedia networks*" by I. Atanasov, E. Pencheva, and D. Marinska, the authors investigate the capabilities of Parlay X Web services for Policy and Charging Control (PCC) in managing all Internet-protocol-based multimedia networks. They explore the requirements for open access to policy and charging control and assess the potential brought by Parlay X Web services compared to PCC. They propose an enhancement to Parlay X Web services to support PCC. The proposed improvement is articulated around the development of new interfaces for usage monitoring, the enhancement of the Quality of Service (QoS) interface, the development of new interfaces for access to QoS-related user data, and the enhancement of call notification functionality. The advantages of these enhancements have been highlighted through the description of case studies.

In the paper, "*Seamless integration of RESTful services into the web of data*", M. Lanthaler and C. Gütl study the seamless integration of RESTful services into semantic web technologies. The proposed idea allows data integration on an unprecedented scale. A new approach (called SEREDASj) is introduced to create machine-readable descriptions for RESTful services as a first step toward solving web development problems. It enables web developers to use tools and knowledge they are already familiar with. In fact, it does not require changes on the described Web services, it provides a viable upgrade path for existing infrastructure. Two algorithms have been proposed to translate SPARQ Update operations to HTTP requests interacting with a SEREDASj-described API. Auch standardized interface increases the developer's productivity and improves code readability.

In the paper, "*A framework for automatic web service discovery based on semantics and NLP techniques*", A. Adala, N. Tabbane, and S. Tabbane propose a combination of semantic and NLP search techniques. This framework is based on natural language processing techniques to match a user request, expressed in natural language, with a semantic web services description. An efficient semantic matching technique has also been proposed to compute the semantic distance between ontological concepts. Compared to existing techniques, the proposed approach exhibits multiple advantages. It offers a simple syntax in terms of a list of keyword phrases and open vocabularies. It maps natural language words into ontological concepts. Furthermore, it does not make any assumption about the description language of the Web service. In fact, it integrates a mapping module which converts English terms from the WordNet database to Suggested Upper Merged Ontology (SUMO).

We hope that this issue will serve as a valuable reference and trend indicator for researchers and engineers in both industry and academia.

Acknowledgments

In closing, we would like to thank all authors for their valuable contributions to this feature topic. We are also grateful to the experts who participated in the review process and completed their reviews within a very tight schedule.

Mohamed Hamdi
Nabil Tabbane
Tai-Hoon Kim
Sajid Hussain

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2012, Article ID 767364, 8 pages
doi:10.1155/2012/767364

Research Article

Image Encryption Using a Lightweight Stream Encryption Algorithm

Saeed Bahrami and Majid Naderi

Cryptography and Secure Systems Laboratory, Faculty of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran, Iran

Correspondence should be addressed to Saeed Bahrami, bahrami.saeed195@gmail.com

Received 2 November 2011; Revised 21 April 2012; Accepted 23 April 2012

Academic Editor: Mohamed Hamdi

Security of the multimedia data including image and video is one of the basic requirements for the telecommunications and computer networks. In this paper, we consider a simple and lightweight stream encryption algorithm for image encryption, and a series of tests are performed to confirm suitability of the described encryption algorithm. These tests include visual test, histogram analysis, information entropy, encryption quality, correlation analysis, differential analysis, and performance analysis. Based on this analysis, it can be concluded that the present algorithm in comparison to A5/1 and W7 stream ciphers has the same security level, is better in terms of the speed of performance, and is used for real-time applications.

1. Introduction

Nowadays, multimedia data such as image and video is expanding in communications and computer networks [1]. Due to widespread use of multimedia data and despite widespread threats and attacks in communication systems, security of this data is necessary [2, 3]. Multimedia encryption challenges originate from two realities. Firstly, multimedia data have great volumes. Secondly, they need real-time uses [4]. So using encryption for security results in additional computations for information processing. As a result, a balance between security and synchronization requirement is necessary [5]. To reach this aim, we use lightweight and high-speed encryption algorithms. One of the methods to ensure security is considering all data as binary strings and encrypt them using block encryption algorithms such as DES. These algorithms are very complex and involve large amounts of computations, and their software implement is not fast enough for high-volume multimedia data [6].

Commonly stream encryption algorithms are used for image encryption [5, 7–9]. Stream ciphers are built using a pseudorandom key sequence, and then this sequence is combined with the original text through exclusive-or operator. Generally, stream encryption systems have suitable performance when speed and error probability of data transmission are high. In this paper, the simple and lightweight stream encryption algorithm is used for multimedia applications such as image, and also various statistical tests are performed in order to assure the security of the algorithm and compared to A5/1 and W7 stream cipher. The notable point in this algorithm is producing the key sequence by AES block cipher in order to enhance the security.

A5/1 and W7 stream cipher algorithms are used for the key production from the linear feedback shift registers. A5/1 algorithm has 64-bit private key, and W7 algorithm has 128-bit private key. Also, both algorithms have adequate security and proper performance speed for image encrypting as compared to block cipher algorithms such as DES, AES, and RC5. Reference [7] provides more details about these two algorithms and their applications in multimedia security.

This paper is classified as follows. In Section 2, one of the stream encryption algorithms is introduced step by step for multimedia use. Section 3 represents a series of security discussion and statistical tests that include visual test, histogram analysis, information entropy, encryption quality, correlation analysis, differential analysis, and performance

analysis introduced and compared to A5/1 and W7 stream cipher. Section 4 concludes the work results.

2. The Stream Encryption Algorithm

As mentioned in the previous section, stream encryption algorithms are used in attention for real-time applications. In this algorithm, stream ciphers are used in order to accelerate implementation of the algorithm. In order to enhance the security, the key product is the same as the key product of AES block cipher.

In this algorithm, the main text is divided in different sections and each section is encrypted by the stream encryption algorithm. In any section, the encryption algorithm uses a separation secret key. The secret key of our encryption schemes is protected by the block cipher (such as AES). $\mathrm{BE}(m, K)$ denotes a block cipher encryption algorithm on message m using key K, and $\mathrm{SE}(P, \mathrm{key}_i)$ denotes a stream cipher encryption algorithm on message P using key key_i. At the beginning of this algorithm, the key of different sections is generated as $\mathrm{key}_i = \mathrm{BE}(m, K)$, then if the plain text is as $P_1, P_2, \ldots, P_t$, the encrypted text would be as $C_1, C_2, \ldots, C_t$, and any section of the encrypted text is as $C_i = \mathrm{SE}(\mathrm{key}_i, M_i)$.

Let F be a function defined as

$$F\left(\mathrm{Key}_i, X\right) = ((((X \times k_1) \oplus k_2) + k_3) \oplus k_4), \tag{1}$$

where Key_i is the 128-bit key and $\mathrm{Key}_i = k_1 k_2 k_3 k_4$ for 32-bits k_i, x is a 32-bit string, $\oplus$ is the bit-wise exclusive-or, $+$ and $\times$ are mod 2^{32} addition and multiplication. To encrypt every 32 bits of the original text, this algorithm has the following steps.

Step 1. A 128-bit key sequence is generated by the block algorithm AES and is considered to be $\mathrm{Key}_i = k_1 k_2 k_3 k_4$ for 32-bit k_i.

It should be noted that this 128-bit key is updated by the AES algorithm to encrypt every 32-bit of the original text.

Step 2. By the function proposed in (1), A_i value is obtained as follows:

$$A_i = F\left(\mathrm{Key}_i, C_{i-1} \oplus P_{i-1}\right), \tag{2}$$

where X value in (1) is replaced by $C_{i-1} \oplus P_{i-1}$. P_{i-1} and C_{i-1} are equal to 32 bits of the previous plain text and cipher text, respectively. In addition, as it was stated above, $\oplus$ is the bitwise exclusive-or.

Step 3. Again, by the function expressed in (1), B_i value is obtained as

$$B_i = F\left(\mathrm{Key}_i, A_i \oplus P_{i-2}\right). \tag{3}$$

In this step, X value in (1) is replaced by $A_i \oplus P_{i-2}$. P_{i-2} is equal to 32 bits of the original text in the two previous cases, and also A_i was obtained in Step 2 by (2).

Step 4. For the third time, (1) is given as

$$D_i = F\left(\mathrm{Key}_i, B_i \oplus C_{i-2}\right). \tag{4}$$

In this equation, X value in (1) is equal to $B_i \oplus C_{i-2}$. C_{i-2} is equal to 32 bits of the encrypted text in the two previous cases, and B_i was obtained in Step 3 by (3).

Step 5. In this stage, according to the following equation, 32 bits of the cipher text are obtained:

$$C_i = P_i \oplus D_i, \tag{5}$$

where P_i value is equal to 32 bits of the plain text and so D_i value was obtained in Step 4.

All the Steps 2–5 can be summarized by the following equation:

$$C_i = P_i \oplus F\left(\mathrm{key}_i, F\left(\mathrm{key}_i, F\left(\mathrm{key}_i, C_{i-1} \oplus P_{i-1}\right) \oplus P_{i-2}\right) \oplus C_{i-2}\right). \tag{6}$$

In all the Steps 2–4, C_0, C_{-1}, p_0, and p_{-1} can be considered equal to k_1, k_2, k_3, and k_4.

The decryption procedure is similar to the encryption one, just with the difference, the locations of P_i and C_i in (5) are exchanged as follows:

$$P_i = C_i \oplus D_i. \tag{7}$$

It should be mentioned that D_i value in the decryption procedure is obtained in accordance with the encryption procedure as well as using the previous original and encrypted texts.

3. Security and Performance Analysis

The main parameter on design of any encryption algorithm is amount of algorithm robustness against cryptographic attacks including brute force, statistical attack, known plain text attack, and chosen plain text attack. Thus, a cipher of high key and plain text sensitivity is desirable. Besides, computational speed and quality of encrypted images are other important issues. In this section, we performed security discussion of the scheme and a series of tests to compare the efficiency of the described algorithm. Images used to implement the tests are some pictures of USC-SIPI image database (freely available at http://sipi.usc.edu/database/).

3.1. Security Discussion of the Scheme

Security of the Key. The key of the encryption/decryption is Key_i that is produced by the BE block cipher. Therefore, achieving the key is difficult.

Meet in the Middle Attack (the Attack to the Section Key). This type of attack is a brute force attack. By meeting one or more bits in the middle, it searches exhaustively the key bits through the middle bits [5]. Since this algorithm has three rounds of F, the meet in the middle attack does not work. Since at least one way to the middle goes through two rounds of F, therefore, the number of key bits that affects a single bit is large.

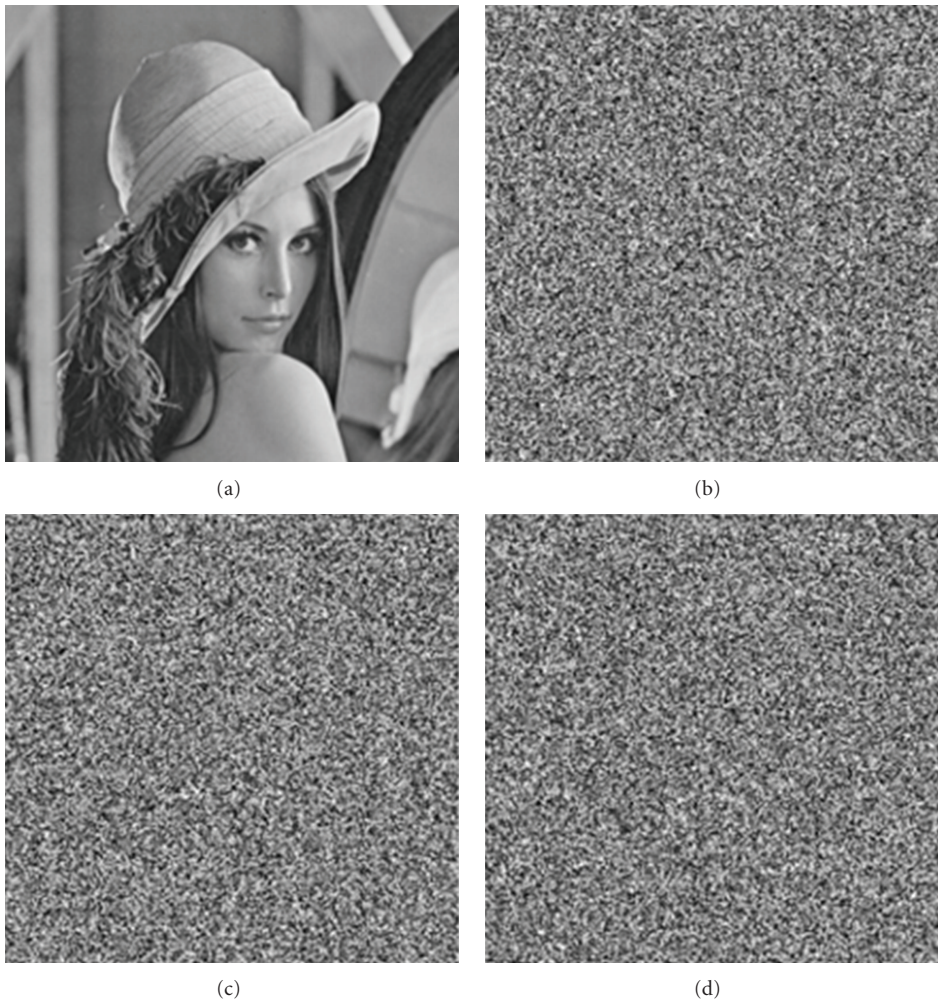

Figure 1: Result of Lena image encryption: (a) the original image, (b) the encrypted image of the described algorithm, (c) the encrypted image of the A5/1 algorithm, (d) the encrypted image of the W7 algorithm.

Chosen Cipher Text Attack (the Attack to the Section Key). All the stream ciphers that have cipher text feedback are weak to the chosen cipher text. For example, if stream cipher was defined as

$$C_i = P_i \oplus F\left(\text{key}_i, F\left(\text{key}_i, F\left(\text{key}_i, C_{i-1}\right) \oplus C_{i-2}\right) \oplus C_{i-3}\right), \tag{8}$$

the cipher would be weak to chose cipher text attack. By choosing $C_{i-3} = C'_{i-3}$, $C_{i-2} = C'_{i-2}$, $C_{i-1} \neq C'_{i-1}$ being different at only one bit, the attacker can ask for the decryption of C_i, C'_i and apply the differential attack [5]. But the stream cipher is defined as

$$C_i = P_i \oplus F\left(\text{key}_i, F\left(\text{key}_i, F\left(\text{key}_i, C_{i-1} \oplus P_{i-1}\right) \oplus b_{i-2}\right) \oplus C_{i-2}\right), \tag{9}$$

where it has both cipher text and plain text feedback. Consequently, achieving the plain text without adequate information from the original text and the encrypted text is impossible.

3.2. Statistical Tests

3.2.1. Visual Test. Observation is an important factor in cipher image test. A good encryption algorithm should mix image so that features are not visually detectable. Also, no information should be observed in the encrypted image by comparing the encrypted and original images [10, 11].

Result of the described algorithm encryption is shown in Figure 1. Figure 1 shows that the encrypted image is quite distinct from the original image.

3.2.2. Histogram Analysis. To prevent the information leakage and aggressive attacks, it must be ensured that the original and encrypted images do not have any statistical similarity. Histogram analysis expresses the way of the distribution of pixels in the image using the drawing number of observations for each amount of pixels brightness [12–16]. Figure 2 shows the histogram analysis on the test image using the described algorithm. The histogram of original image has a sharp rise with a sharp decline as shown in Figure 2(a), and histogram of the encrypted image as shown in Figure 2(b)

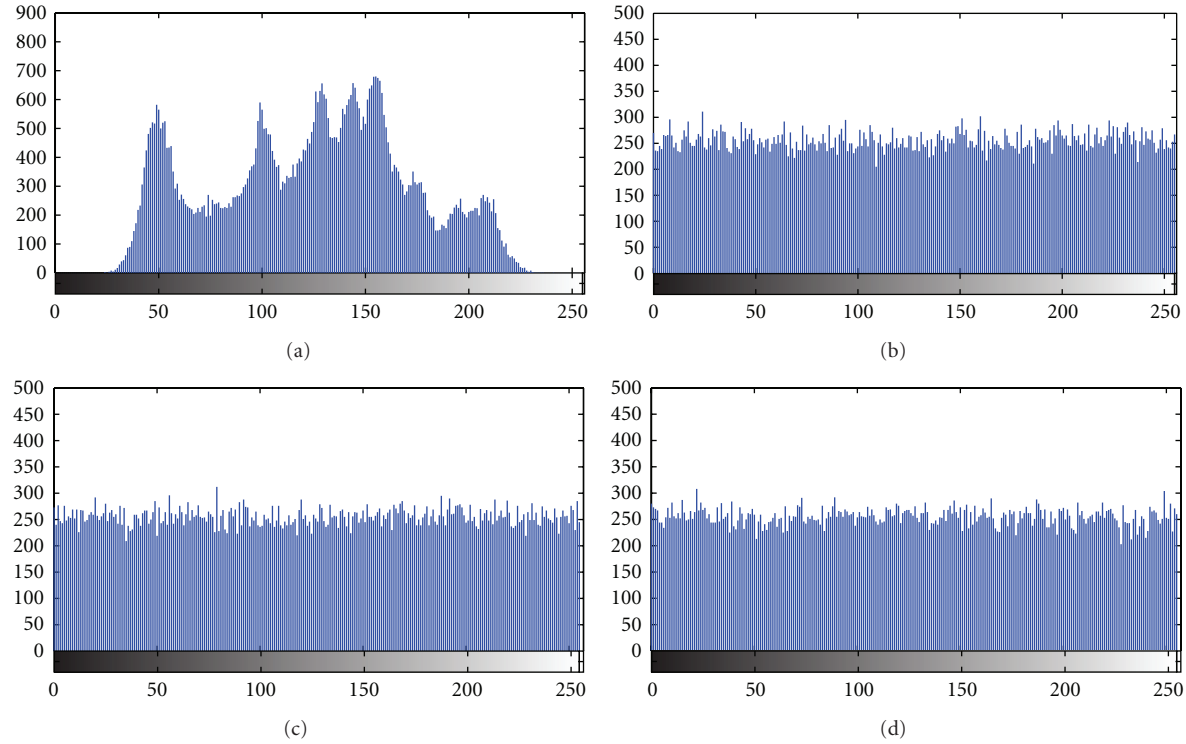

Figure 2: Result of histogram analysis. (a) Histogram of the original image, (b) the described algorithm, (c) A5/1 algorithm, and (d) W7 algorithm.

has a uniform distribution that is completely different from histogram of the original image and has no statistical similarity. Therefore, the attacker with the histogram analysis of the encrypted image cannot acquire information from the original image.

3.2.3. Information Entropy. Shannon introduced information entropy as the measure of source information in 1949. The $H(s)$ entropy of a message source s is defined as

$$H(s) = \sum_{i=0}^{2N-1} P(s_i)\log_2 \frac{1}{P(s_i)}. \quad (10)$$

In this equation, $P(s_i)$ represents the probability of symbol s_i and the entropy is expressed in bits [17]. If we suppose that the source emits 2^8 symbols with equal probability and $s = \{s_1, s_2, \ldots, s_{2^8}\}$, random source entropy is equal to 8. If an encryption algorithm creates symbols with entropy less than 8, there is likelihood to predict original image from encrypted image, which is a threat to the system security. As it is observed in Table 1, entropy of studied algorithms is very close to the ideal value of 8. This means that information leakage in the encryption process is negligible and studied algorithms are secure upon the entropy attack. Also, we conclude that the entropies of A5/1 and the proposed algorithms are closer to the ideal value compared with entropy of W7.

3.2.4. Encryption Quality. The image encryption creates large changes in the amount of pixels. These pixels are completely different from the original image. These changes are irregular. More changes in values of the pixels show more effectiveness of encryption algorithm and thus better quality. Let $C(x, y)$ and $P(x, y)$ be the gray level of the pixels at the xth row and yth column of a $W \times S$ encrypted and original images, respectively. Encryption quality shows the average of changes in each amount of gray L, and, according to [18], it can be expressed as

$$\text{Encryption Quality} = \frac{\sum_{L=0}^{255} |H_L(C) - H_L(P)|}{256}, \quad (11)$$

where $H_L(P)$ and $H_L(C)$ are the number of repetition from each gray value in the original image and the encrypted

Table 1: Entropy results of encrypted images. Grayscale type with 256×256 size.

File name	File description	The proposed algorithm	A5/1	W7
4.2.04	Girl (Lena)	7.9890	7.9892	7.9886
5.1.12	Clock	7.9899	7.9901	7.9893
5.1.13	Resolution chart	7.9870	7.9890	7.9869
5.1.14	Chemical plant	7.9894	7.9899	7.9894
5.2.08	Couple	7.9884	7.9897	7.9885
5.2.09	Aerial	7.9897	7.9898	7.9897
5.2.10	Stream and bridge	7.9894	7.9896	7.9893
5.3.01	Man	7.9891	7.9899	7.9889
5.3.02	Airport	7.9898	7.9898	7.9899

Table 2: Quality results of encrypted images. Grayscale type with 256×256 size.

File name	File description	The proposed algorithm	A5/1	W7
4.2.04	Girl (Lena)	170	169.38	168.50
5.1.12	Clock	242.80	242.33	241.59
5.1.13	Resolution chart	454.91	455.33	454.61
5.1.14	Chemical plant	206.33	207.10	206.14
5.2.08	Couple	235.55	222.86	220.66
5.2.09	Aerial	265.77	267.99	265.14
5.2.10	Stream and bridge	140.99	141.32	140.84
5.3.01	Man	145.16	145.09	143.71
5.3.02	Airport	288.84	289.16	288.08

image, respectively. Encryption quality for A5/1, W7, and the described algorithm is available for different images in Table 2. From the obtained values, we conclude which the qualities of A5/1 and the proposed algorithm are better than W7.

3.2.5. Correlation Analysis. Any pixel correlates highly with adjacent pixels in the original image. Equations (5), (6), and (7) are used to study the correlation between adjacent pixels in horizontal, vertical, and diagonal orientations [4, 7, 15, 16]:

$$r_{xy} = \frac{\mathrm{Cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}},$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_j - \frac{1}{N}\sum_{i=1}^{N}x_j\right)^2, \tag{12}$$

$$\mathrm{Cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}\left(x_j - \frac{1}{N}\sum_{i=1}^{N}x_j\right)\left(y_j - \frac{1}{N}\sum_{i=1}^{N}y_j\right).$$

In these equations, r_{xy} is correlation coefficient, x and y are intensity values of two adjacent pixels in the image, and N is the number of pair pixels of the selected adjacency in the image to calculate the correlation. 1000 pairs of two adjacent pixels are selected randomly from the image. Ideally, correlation coefficient of the original image is equal to one, and the correlation coefficient of the encrypted image is equal to zero. Also, the correlation diagram is used. Initially, the neighborhood of horizontal, vertical, and diagonal of N pixels is identified in this diagram. Then, diagram is plotted based on the value of each pixel and its neighbors.

As it is specified in Figure 3, correlation between pixels of the original image is too much, while there is a little correlation between neighboring pixels in the encrypted image. In Table 3, correlation coefficients of different encrypted images by studied encryption algorithms have been given for neighborhoods of horizontal, vertical, and diagonal. The table shows that the values of correlation coefficients of the three algorithms are very close to zero for each neighborhood. Therefore, these algorithms are secure against correlation attacks.

3.2.6. Differential Analysis. An encryption algorithm should be designed so that it is sensitive to the small changes in the original image. Attacker tries to view the changes result in the encrypted image making minor changes in the original image. Thus, it reveals a significant relationship between the original image and the encrypted image. Also, this action facilitates finding the algorithm key. If a small change in the original image can cause a large change in the encrypted image, then the differential attack is not possible.

Three common measures were used for differential analysis: MAE, NPCR, and UACI [7, 17]. MAE is mean absolute error. NPCR is the number of pixels change rate of encrypted image, while one pixel of original image is changed.

UACI is the unified average changing intensity, which measures the average intensity of the differences between the original image and the encrypted image.

If $C(x,y)$ and $P(x,y)$ are the gray level of the pixels at the xth row and yth column of a $W \times S$ encrypted and original image, respectively, then MAE is defined as

$$\mathrm{MAE} = \frac{1}{H \times W}\sum_{x=0}^{H-1}\sum_{y=0}^{W-1}|\,C(x,y) - P(x,y)\,|. \tag{13}$$

The MAE test results for the three encryption algorithms have been recorded in Table 4. Information recorded in the table shows that the calculated MAE values of encryption algorithms have little difference.

Consider two encrypted images C_k and $\bar{C}_k$ that, corresponding to original images, are only different in a pixel. The NPCR is defined as

$$\mathrm{NPCR}_k = \frac{\sum_{x=0}^{H-1}\sum_{y=0}^{W-1}D_k(x,y)}{H \times W} \times 100\%,$$
$$D_k(x,y) = \begin{cases} 0, & C_k(x,y) = \bar{C}_k(x,y), \\ 1, & C_k(x,y) \neq \bar{C}_k(x,y), \end{cases} \tag{14}$$

and UACI is defined as

$$\mathrm{UACI}_k = \frac{1}{H \times W} \times \sum_{x=0}^{H-1}\sum_{y=0}^{W-1}\left[\frac{\left|C_k(x,y) - \bar{C}_k(x,y)\right|}{255}\right] \times 100\%. \tag{15}$$

It is clear that large amounts of NPCR and UACI indicate a high sensitivity of the encryption algorithm to the original image. The NPCR and UACI test results have been recorded in Table 5. The results indicate that the NPCR and UACI are less than 0.01% for the studied algorithms. Unfortunately, this means that these algorithms have low sensitivity to changes in the original image.

3.2.7. Performance Analysis. In addition to security issues, the speed of encryption algorithm is important for real-time processing. Efficiency of the proposed encryption algorithm is dependent on the comparison between the speed of encryption algorithms. Efficiency of algorithms has been

Table 3: Correlation coefficient results of encrypted images. Grayscale type with 256 × 256 size.

File name	File description	Neighborhood of horizontal			Neighborhood of vertical			Neighborhood of diagonal		
		The proposed algorithm	A5/1	W7	The proposed algorithm	A5/1	W7	The proposed algorithm	A5/1	W7
4.2.04	Girl (Lena)	−0.0074	−0.0072	−0.0012	0.0072	−0.0522	−0.0122	0.0105	0.0131	0.0017
5.1.12	Clock	0.0320	−0.0130	0.0236	0.0068	−0.0230	0.0220	−0.0840	0.0015	0.0057
5.1.13	Resolution chart	−0.0042	−0.0311	0.0076	0.0196	0.0180	−0.0033	0.0166	−0.0064	0.0196
5.1.14	Chemical plant	−0.0132	0.0177	0.0221	0.0186	−0.0165	0.0364	0.0162	0.0038	−0.0099
5.2.08	Couple	−0.0048	0.0194	0.0227	−0.0149	0.0322	0.0205	0.0149	−0.0052	0.0131
5.2.09	Aerial	0.0094	0.0053	0.0083	−0.0218	0.0098	0.0178	0.0116	−0.0128	0.0184
5.2.10	Stream and bridge	−0.0015	0.0196	0.0017	−0.0357	0.0234	−0.0194	−0.0114	0.0067	−0.0034
5.3.01	Man	−0.0144	0.0084	−0.0402	0.0285	−0.0059	0.0099	−0.0067	0.0085	0.0175
5.3.02	Airport	0.0088	0.0131	0.0166	0.0029	−0.0153	0.0179	0.0196	−0.0223	−0.0182

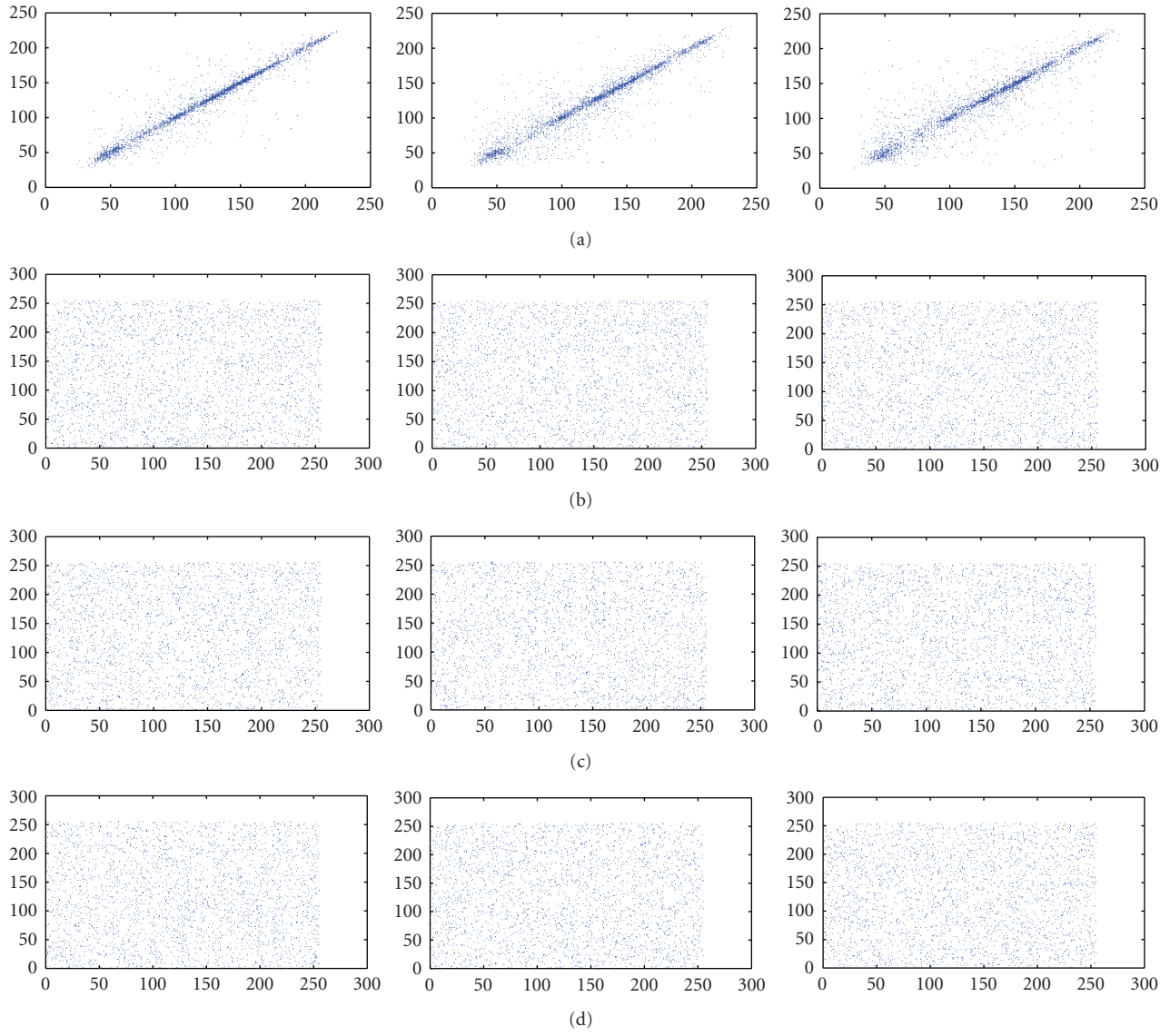

Figure 3: Correlation chart. left side with neighborhood of horizontal, center with neighborhood of vertical, right side with neighborhood of diagonal (a) Lena Standard image (b) the described algorithm (c) A5/1 algorithm and (d) W7 algorithm.

Table 4: MAE test results for Lena standard.

Image size	type	The proposed image	A5/1	W7
256 × 256	grey	72.969	72.706	72.62

Table 5: Compare UACI and NPCR between encryption algorithms in Lena standard image.

Method	NPCR	UACI
The proposed algorithm	%0.0702	%0.0262
A5/1	%0.0015	%0.0005
W7	%0.0015	%0.0006

Table 6: Compare the speed of the studied algorithms in MATLAB programming environment.

Encryption speed comparison (second)			
Size	64 × 64	128 × 128	256 × 256
AES-128	4.23	10.12	65.23
The proposed algorithm	0.191	0.51	1.21
A5/1	0.21	0.87	2.32
W7	0.42	1.01	3.23

achieved with a unoptimized MATLAB code on a machine with Intel core 2 Duo 2.10 processor and 2 Gbytes of RAM memory for Windows 7 operating system. The results in Table 6 show that the described algorithm in terms of execution speed is better than algorithms A5/1 and W7 and so is better for real-time applications.

4. Conclusion

In this investigation, one stream encryption algorithm was proposed for multimedia systems, and many statistical tests were performed to prove suitability of the algorithm, and so this algorithm was compared to A5/1 and W7 stream ciphers. Based on the visual test, there is not any kind of information from the original image in the encrypted image. The histogram shows that distribution of brightness in pixels of the encrypted image is completely uniform, and there is not any statistical similarity with the histogram of the original image. The results of information entropy test show that this value is very close to the ideal value in the encrypted images for all three algorithms. Consequently, these algorithms are secure against entropy attacks. Also, comparison between the entropy of the three algorithms shows that entropies of A5/1 and the proposed algorithms are closer to the ideal value compared with entropy of W7. Based on the results of the encryption quality, the described and A5/1 algorithms have a better quality in the diffusion and confusion of pixels than W7 algorithm. Diagram and coefficients of correlation show that correlation between pixels of the encrypted image has declined severely, and these algorithms are secure against correlation attacks. In order to measure the sensitivity of the algorithm to minor changes in the original image, two measures were considered: NPCR and UACI. The results showed that the proposed algorithm and A5/1 and W7 algorithms have a little sensitivity to minor changes in the original image, ultimately. Performance speed of the described algorithm and two algorithms of A5/1 and W7 were compared. The results showed that performance speed of the described algorithm is faster than two algorithms of A5/1 and W7. According to last discussions, it seems that the described algorithm in software applications has more advantages compared to both algorithms of A5/1 and W7.

References

[1] A. Uhl and A. Pommer, "Application scenarios for the encryption of still visual data," in *Image and video encryption from Digital Rights Management to secured personal communication, Advances in Information Security*, vol. 15, pp. 31–43, Springer, 2005.

[2] S. Lian and X. Chen, "On the design of partial encryption scheme for multimedia content," *Mathematical and Computer Modelling*. In press.

[3] N. Taneja, B. Raman, and I. Gupta, "Combinational domain encryption for still visual data," *Multimedia Tools and Applications*, vol. 59, no. 3, pp. 775–793, 2012.

[4] S. S. Agaian, R. G. R. Rudraraju, and R. C. Cherukuri, "Logical transform based encryption for multimedia systems," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '10)*, pp. 1953–1957, October 2010.

[5] F. Bao and R. H. Deng, "Light-weight encryption schemes for multimedia data and high-speed networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 188–192, November 2007.

[6] C. Li, S. Li, M. Asim, J. Nunez, G. Alvarez, and G. Chen, "On the security defects of an image encryption scheme," *Image and Vision Computing*, vol. 27, no. 9, pp. 1371–1381, 2009.

[7] A. Jolfaei and A. Mirghadri, "Survey: image Encryption Using A5/1 and W7," vol. 2, no. 8.

[8] N. Thomas, D. Redmill, and D. Bull, "Secure transcoders for single layer video data," *Signal Processing*, vol. 25, no. 3, pp. 196–207, 2010.

[9] F. Liu and H. Koenig, "A survey of video encryption algorithms," *Computers and Security*, vol. 29, no. 1, pp. 3–15, 2010.

[10] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[11] A. Pande and J. Zambreno, "The secure wavelet transform," *Journal of Real-Time Image Processing*, vol. 18, no. 3, pp. 844–856, 2010.

[12] C. N. Raju, G. Umadevi, K. Srinathan, and C. V. Jawahar, "Fast and secure real-time video encryption," in *Proceedings of the 6th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP '08)*, pp. 257–264, December 2008.

[13] J. Zhou, Z. Liang, Y. Chen, and O. C. Au, "Security analysis of multimedia encryption schemes based on multiple Huffman table," *IEEE Signal Processing Letters*, vol. 14, no. 3, pp. 201–204, 2007.

[14] W. Li and N. Yu, "A robust chaos-based image encryption scheme," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '09)*, pp. 1034–1037, July 2009.

[15] R. C. Luo, L. Y. Chung, and C. H. Lien, "A novel symmetric cryptography based on the hybrid haar wavelets encoder and chaotic masking scheme," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 4, pp. 933–944, 2002.

[16] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons and Fractals*, vol. 21, no. 3, pp. 749–761, 2004.
[17] C. E. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal*, vol. 28, pp. 656–715, 1949.
[18] H. E. D. H. Ahmed, H. M. Kalash, and O. S. Farag Allah, "Encryption quality analysis of the RC5 block cipher algorithm for digital images," *Optical Engineering*, vol. 45, no. 10, Article ID 107003, 2006.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2012, Article ID 630610, 9 pages
doi:10.1155/2012/630610

Research Article

A Novel *k*-out-of-*n* Oblivious Transfer Protocol from Bilinear Pairing

Jue-Sam Chou

Department of Information Management, Nanhua University, No. 55, Section 1, Nanhua Road, Dalin Township, Chiayi County 62249, Taiwan

Correspondence should be addressed to Jue-Sam Chou, jschou@mail.nhu.edu.tw

Received 30 November 2011; Revised 13 March 2012; Accepted 27 March 2012

Academic Editor: Mohamed Hamdi

Oblivious transfer (OT) protocols mainly contain three categories: 1-out-of-2 OT, 1-out-of-*n* OT, and *k*-out-of-*n* OT. In most cases, they are treated as cryptographic primitives and are usually executed without consideration of possible attacks that might frequently occur in an open network, such as an impersonation, replaying, or man-in-the-middle attack. Therefore, when used in certain applications, such as mental poker games and fair contract signings, some extra mechanisms must be combined to ensure the security of the protocol. However, after a combination, we found that very few of the resulting schemes are efficient enough in terms of communicational cost, which is a significant concern for generic commercial transactions. Therefore, we propose a novel *k*-out-of-*n* oblivious transfer protocol based on bilinear pairing, which not only satisfies the requirements of a *k*-out-of-*n* OT protocol, but also provides mutual authentication to resist malicious attacks. Meanwhile, it is efficient in terms of communication cost.

1. Introduction

An oblivious transfer (OT) is an important primitive for designing security services. It can be used in various applications like the signing of fair contracts, oblivious database searches, mental poker games, privacy-preserving auctions, secure multiparty computations [1], and so on. In 1981, Rabin [2] first proposed an interactive OT scheme in which the probability of the receiver's capability to decrypt a message sent by the sender is 1/2. Rabin used the proposed OT to design a 3-pass secret exchange (EOS) protocol, hoping that two parties can exchange their secrets fairly. In 1985, Even et al. [3] presented a more generalized OT, called 1-out-of-2 OT (OT_1^2), in which a sender sends two encrypted messages to a chooser with only one of which the chooser can decrypt. They also presented a contract-signing protocol by evoking OT_1^2 multiple times to prevent one party from obtaining the other party's contract signature without first showing his own. In 1986, Brassard et al. [4] further extended OT_1^2 into a 1-out-of-*n* OT (OT_1^n, also known as "all-or-nothing"), in which only one out of *n* sent messages can actually be obtained by the chooser. The authors pointed out that their OT_1^n scheme can be used to implement a multiparty mental poker game [5] against a player coalition. In contrast to the interactive versions described above, Bellare and Micali [6] first proposed a noninteractive OT_1^2 scheme in 1989. In this scheme, a user obliviously transfers two messages to another party equipped with two public keys to decrypt one of the messages.

From 1999 to 2001, based on the above-mentioned interactive and noninteractive OT schemes, Naor and Pinkas proposed some related OT methods, such as an adaptive OT_k^n [7], proxy OT_1^2 [8], distributed OT_k^n [9], efficient OT_1^n [10], and efficient OT_k^n [11]. Here, OT_k^n is the final form of the OT schemes. In this form, from the *n* encrypted messages sent, the chooser can obtain *k* chosen messages in plaintext form without the sender's knowledge regarding which part of the messages are decrypted. In Naor and Pinkas's distributed OT_k^n schemes [9], the sender distributes two messages (M_0, M_1) among *n* servers, and the chooser contacts k $(k < n)$ servers to receive one $(M_\sigma, \sigma = 0 \text{ or } 1)$ of them. The authors claimed that their schemes can protect the privacy of both parties. However, in 2007, Ghodosi [12] showed two possible attacks on these schemes. In

the first attack, two collaborating servers can reveal the chooser's choice of σ, while, in the second attack, the chooser can learn both M_0 and M_1 by colluding with only a single server. In 2002, Mu et al. [13] proposed three OT_k^n schemes constructed using RSA encryption, a Nyberg-Rueppel signature, and an ElGamal encryption scheme, respectively. Two of these are interactive, while the other can be either interactive or noninteractive. The authors claimed that their schemes are complete, robust, and flexible and induce a significant improvement in communication cost. However, in 2006, Ghodosi and Zaare-Nahandi [14] showed that these schemes fail to satisfy the requirements of an oblivious transfer protocol. In 2004, Ogata and Kurosawa [15] proposed another OT_k^n scheme, based on an RSA blind signature, which can be employed in either an adaptive or a nonadaptive manner. The authors claimed that their scheme can be applied to oblivious key searching. In 2005, three OT_k^n schemes are proposed [16–18]. Among these, Chu and Tzeng's scheme [16] is the most efficient as it needs only 2 passes to send 1024 k bits from the chooser to the sender, and $1024^*(k+1) + n^*|\mathrm{Data}|$ bits from the sender to the chooser, where Data is a message or ciphertext, and $|\mathrm{Data}|$ represents the bit length of Data. In 2006, Parakh [19] proposed an elliptic-curve-based algorithm allowing A to obliviously transfer his secrecy, n_A, to B with a 50% probability of success. However, we found that A can decide whether B can obtain his secret n_A (which is one-to-one mapped to Pn_A) by first assuming that $P_A = P_B$. Under this assumption, upon receiving $\{n_BP_B; n_B(n_AP_A) + R; n_BR\}$ from B, A can obtain B's one-time random variable R by computing $(n_B(n_AP_A) + R) - n_A(n_BP_B)$. Then, by computing $n_A(n_BR) = n_B(n_AR)$, A can obtain n_BK. Subsequently, by computing $(n_A(n_B\ R) + Pn_A) - n_BK$, A obtains Z_B, just as B does in step 5(b). Therefore, if A finds $Z_B = Pn_A$, it confirms that B can obtain n_A after the protocol runs; otherwise, it knows B cannot obtain the value of n_A. This violates B's privacy. In the same year, for coping with all possible attacks encountered in an open network, Kim and Lee [20] proposed two OT_1^2 protocols, which are modified from Bellare-Micali noninteractive OT_1^2 scheme [6] by appending the sender's signature to make the sender undeniable about what he sent and be authentic to the chooser. However, we found, other than the weaknesses pointed by Chang and Shiao [21], Kohnfelder's protocol still has the reblocking problem [22]. Because when modulus $n_A > n_B$, message M_A cannot be recovered by Bob. This makes legal Alice unable to be authenticated by Bob.

In 2007, Halevi and Kalai [23] proposed another OT_1^2 scheme by using smooth projective hashing and showed that the used RSA composite in their scheme need not be a product of safe primes. Also in 2007, Camenish et al. and Green and Hohenberger proposed two related OT schemes [24, 25], respectively. Both focus on the security of full simulatability for the sender and receiver to resist against selective-failure attack [7]. In 2009, Qin et al. [26] proposed two noninteractive OT_1^n schemes. However, in their protocols, a receiver has to interact with a third party to obtain the choice-related secret key each time it wants to select one of the n sent message. This makes their scheme somewhat inconvenient and inconsistent with the meaning of noninteractive protocols as indicated in the title (this phenomenon can also be found in some proposed noninteractive OT schemes). In the same year, Chang and Lee [27] presented a robust OT_k^n scheme using both the RSA blind signature and Chinese Remainder Theorem. However, we found their scheme fails since the sender can decide which parts of the messages were chosen by the chooser. We will describe this weakness in Section 3.2. In addition, in 2011, Ma et al. [28] proposed an oblivious transfer using a privacy scheme for a timed-release receiver. Their scheme has a good timed-release property. However, it needs to call ZKP k times to learn k of the n sent messages. This makes their protocol less efficient. Moreover, it does not have mutual authentication. Therefore, when the sender and receiver want to communicate, they need a secure channel. Otherwise, without identity authentication, malicious attackers can simultaneously launch many ZKPs. This will degrade the system performance and may cause the system to suffer from a denial-of-service (DOS) attack (according to the definition in [29]).

After surveying all of the above-mentioned OT schemes, we found that almost all of them lack the consideration of adding security features. Only [2, 20] do consider the protection against all possible attacks. However, study [20] fails which we have described earlier. Hence, if we wish all of the proposed OT protocols, other than scheme [2], to be able to resist against various attacks, we should run them through secure channels. This would incur extra communicational overhead. For this reason, in this paper, we propose a novel interactive OT_k^n scheme that needs only two passes but can get rid of using a secure channel to avoid adding extra communicational overhead. It not only is simple in concept but also encompasses some essential security features such as mutual authentication, the prevention of man-in-the-middle (MIMA) attack, and replay attack. Thus, when compared with other interactive OT schemes, our scheme promotes not only in the communicational efficiency but also in the aspect of security.

The rest of this paper is organized as follows. The introduction has been presented in Section 1, and some preliminaries are shown in Section 2. In Section 3, we review Chang et al.'s scheme and show its weakness. After that, we show our protocol in Section 4. Then, the security analyses and communicational cost comparisons among related works and our scheme are made in Section 5. Finally, a conclusion is given in Section 6.

2. Preliminaries

In this section, we briefly introduce the security features of our OT_k^n scheme in Section 2.1, the principles of bilinear pairing in Section 2.2, and some intractable problems used in this paper in Section 2.3.

2.1. Security Features of Our OT_k^n Scheme. Just as traditional OT schemes, our OT_k^n also has two parties, the sender S and the chooser C. In the scheme, S obliviously transfers

n messages to C, and C can choose k messages among them without S's knowledge about which k messages were selected, where $n \geq 2$ and $k < n$. In addition, our scheme also possesses the following three security features which are needed in a traditional OT scheme.

(1) Correctness. After the protocol run, C should be able to obtain the valid data chosen by him before.

(2) Chooser's Privacy. In the protocol, each of the k chooser's choices should not be known to the sender or any third party. More precisely, each of the chooser's encrypted choice can be any valid choice with equal probability, that is, for an encrypted choice y and any valid choice x, $\Pr[x \mid y] = \Pr[x]$. This property is known as *Shannon perfect secrecy*.

(3) Sender's Privacy. At end of the protocol run, the chooser cannot get any knowledge about the other messages it did not choose. More formally, the ciphertexts sent by the sender are semantically secure [30]. The chooser can obtain a plaintext decrypted from its ciphertext only if it has the key offered by the sender.

Except for the above three properties, our interactive OT_k^n scheme also has the following three security features, (4) through (6), to guard against possible security threats.

(4) Impersonation Attack Resistance. Each party has to authenticate the counterpart. That is, it should be a mutual-authentication OT.

(5) Replaying Attack Resistance. An adversary could not obtain any messages by only replaying old messages sent by the sender.

(6) Man-in-the-Middle Attack (MIMA) Resistance. MIMA is an attack that an adversary eavesdropping on the communication line between two communicating parties uses as some means to make them believe that they each are talking to the intended party. But indeed, they are talking to the adversary.

2.2. Bilinear Pairing. Let G_1 be an additive group composed of points on an elliptic curve with order q, and let G_2 be a multiplicative group with the same order. A bilinear mapping is defined as $\hat{e} : G_1 \times G_1 \rightarrow G_2$ which must satisfy the following properties [31].

(1) Bilinear: a mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is bilinear if $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and all $a, b \in Z_q^*$.

(2) Nondegenerate: the mapping does not map all pairs in $G_1 \times G_1$ to the identity in G_2.

(3) Computable: there is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in G_1$.

(4) If P is a generator for G_1 then $\hat{e}(P, P)$ is a generator for G_2.

(5) Commutative: for all $P_1, P_2 \in G_1$, $\hat{e}(P_1, P_2) = \hat{e}(P_2, P_1)$.

(6) Distributive: for all $P_1, P_2, P_3 \in G_1$, $\hat{e}(P_1 + P_2, P_3) = \hat{e}(P_1, P_3)\hat{e}(P_2, P_3)$.

2.3. Some Diffie-Hellman Problems. Let $a, b, c, g \in_R Z_q^*$, let P be a base point of a group on an Elliptic curve, and let $G = \langle g \rangle$, $G_1 = \langle P \rangle$, and $G_2 = \langle g (= \hat{e}(P, P)) \rangle$ *be* three groups with each having a prime order q. Using these definitions, we describe some well-known intractable Diffie-Hellman problems [32] that will be used in this paper.

(1) The Computational Diffie-Hellman (CDH) Problem. In G, given (g, g^a, g^b), finding the element $C = g^{ab} \bmod q$.

(2) The Decisional Diffie-Hellman (DDH) Problem. In G, given (g, g^a, g^b, g^c), deciding whether $c = ab \bmod q$.

(3) The Bilinear Computational Diffie-Hellman (BCDH) Problem. Given (P, aP, bP, cP) in G_1, finding $\hat{e}(P, P)^{abc}$ in G_2.

According to Boneh and Franklin's study [31], the BCDH problem is no harder than the CDH problem in G (or equivalently G_2).

(4) Chosen-Target CDH (CTCDH) Problem. Let $H : \{0,1\}^* \rightarrow G$ be a hash function, let $T(\cdot)$ be a target oracle which returns a random element in G, and $(\cdot)^c$ a helper oracle which returns $T(j)^c$ when queried by $T(j)$, where c is an unknown random integer in Z_q^*. Also, let q_t be the number of queries to $T(\cdot)$ and q_h the number of queries to $(\cdot)^c$. The CTCDH problem is finding l pairs of $(j_1, \nu_1), \ldots,$ and (j_l, ν_l), with each satisfying $\nu_i = (T(j_i))^c$, for $1 \leq i \leq l$ and $q_h < l \leq q_t$. Without loss of generality, we can let q_h and q_t be $l - 1$ and l, respectively. The CTCDH problem can then be rephrased as that after obtaining $T(j_1), \ldots, T(j_l)$ and $(j_1, \nu_1), \ldots, (j_{l-1}, \nu_{l-1})$ via querying the $T(\cdot)$ oracle and the helper oracle $(\cdot)^c$ correspondingly, trying to find the lth pair (j_l, ν_l) without the knowledge of c. The CTCDH problem is proposed and considered as a hard problem by Boldyreva in 2002 [33]. Its former version in RSA is proved by Bellare et al. in [34].

3. Review of Chang et al.'s Protocol

In 2009, Chang et al. proposed a robust OT_k^n scheme based on CRT, hoping that their scheme can achieve the security requirements of a general OT_k^n scheme. However, we found their scheme cannot satisfy the chooser's privacy. In the following, we first review the scheme in Section 3.1 then show the weakness found in Section 3.2.

3.1. Review. We roughly describe the protocol by listing the relevant steps in the following (see [27] for more details).

Step 1. After receiving the request from Bob for all messages $a_1, a_2, \ldots, a_n$, Alice owning these n messages selects n

relatively prime integers, $d_1, d_2, \ldots, d_n$, and computes $D = d_1 * d_2 * \cdots * d_n$. She then constructs the congruence system

$$C \equiv a_1 \bmod d_1,\ C \equiv a_2 \bmod d_2, \ldots, \quad C \equiv a_n \bmod d_n. \tag{1}$$

Furthermore, Alice computes the following values: $T_1 = d_1^e \bmod N$, $T_2 = d_2^e \bmod N, \ldots$, and $T_n = d_n^e \bmod N$, where N be the product of two large primes and (e, d) be Alice public/private key pair satisfying $ed = 1 \bmod \varphi(N)$, by using her public key e. Finally, she publishes C and the n pairs of (ID_i, T_i), for $i = 1$ to n, in the public board.

Step 2. If Bob wants to learn k messages among them, he must select k pairs of (ID'_j, T'_j), for $j = 1$ to k, from the public board and first generate k corresponding random numbers $r_1, r_2, \ldots, r_k$, for each pair of (ID'_j, T'_j). Then, he subsequently computes the following:

$$\alpha_1 = r_1^e * T'_1 \bmod N,\ \alpha_2 = r_2^e * T'_2, \quad \bmod N, \ldots, \alpha_k = r_k^e * T'_k \bmod N, \tag{2}$$

by using Alice's public key e and sends $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ back to Alice.

Step 3. Upon receiving the messages sent by Bob, Alice employs her private key d to compute $\beta_1 = \alpha_1^d = r_1 T_1'^d = r_1 d'_1 \bmod N$, $\beta_2 = \alpha_2^d = r_2 T_2'^d = r_2 d'_2 \bmod N, \ldots, \beta_k = \alpha_k^d = r_k T_k'^d = r_k d'_k \bmod N$ and then sends the results $\{\beta_1, \beta_2, \ldots, \beta_k\}$ to Bob.

Step 4. After receiving the messages from Alice, Bob computes the following values: $d'_1 = r_1^{-1} * \beta_1 \bmod N, d'_2 = r_2^{-1} * \beta_2 \bmod N, d'_k = r_k^{-1} * \beta_k \bmod N$. Consequently, Bob learns the demanded messages successfully by computing

$$b_1 = C \bmod d'_1,\ b_2 = C \bmod d'_2, \ldots, \quad b_k = C \bmod d'_k. \tag{3}$$

3.2. Weaknesses. Although Chang et al. claimed that their scheme can satisfy the security requirements demanded by the OT_k^n scheme, we found that Bob's privacy has been violated, since according to their protocol, Alice first sets n values of d_i ($i = 1$ to n), and Bob commits his k choices to the k values of α_j ($j = 1$ to k). After computing the k values of β_j ($= 1$ to k), Alice can use each of the $d_i^{-1}s$ ($i = 1$ to n) to compute $r_{ji} = \beta_j * d_i^{-1}$, for $j = 1$ to k and $i = 1$ to n. In addition, using each r_{ji}, Alice can compute the n values of $\alpha_i^{(*)} = (r_{ji} * d_i)^e$, for $i = 1$ to n, to compare with the k committed values, α_j. For example, suppose Bob chooses the first message, $T_1 = {d_1}^e \bmod N$, and Alice wants to guess which T_i Bob chose, Alice starts to use ${d_1}^{-1}$ to compute $r_{11} = \beta_1 * {d_1}^{-1} \bmod N = {\alpha_1}^d (= r_1 * d_1) * {d_1}^{-1} \bmod N = r_1 \bmod N$. He will get $\alpha_1{}^{(*)} = (r_{11} * d_1)^e \bmod N = \alpha_1 = r_1^e * T_1$. That is, Alice will find a match, α_1, and knows that Bob chose the first message. Conversely, if Alice uses d_i^{-1}, $(i = 2, n)$ to compute $r_{1i} = \beta_1 * d_i^{-1}$, he will get $\alpha_i^{(*)} = (r_{1i} * d_i)^e \bmod N$, which is not equal to α_1. In other words, Alice cannot know the correct message T_1 that Bob chose. That is, once a pair, $(\alpha_i{}^{(*)}, \alpha_i)$, for example, has been matched, Alice knows that Bob chose the ith message. Hence, we can easily see that such explorations cost at most $n * k$ multiplications to obtain r_{ji}, and $n^2 * k$ multiplications and $n^2 * k$ exponentiations to yield all values of $\alpha_i^{(*)}$. Therefore, with at most $(n^2 * k + n * k)$ multiplications and $n^2 * k$ exponentiations, it is computationally feasible for Alice to decide which k values Bob selected, which violates Bob's privacy.

4. Proposed Protocol

In this section, we present our ID-based OT_k^n protocol based on bilinear pairings, which were proved and applied to cryptography by Boneh and Franklin in 2001 [31]. Our scheme consists of two phases: (1) an initialization phase and (2) an oblivious transfer phase. In the following, we first describe these two phases. Then, to demonstrate the chooser's privacy preservation, we use a misleading attack for an explanation. As the receiver's privacy preservation can be reasoned in a similar fashion, we omit its description here.

(1) Initialization Phase. In this phase, we adopt the same system parameters as the ones used in [31]. In addition, there also exists a trusted key generation center (KGC) which is assumed to be key-escrow-attack free. Initially, KGC chooses an additive group $G_1 = \langle P \rangle$ of order q, a multiplicative group $G_2 = \langle \hat{e}(P, P) \rangle$ of the same order, where $\hat{e}$ is a bilinear mapping, that is, $\hat{e} : G_1 \times G_1 \to G_2$, and three one-way hash functions: H: $\{0,1\}^* \to \{0,1\}^l$, $H_2 : G_1 \to \{0,1\}^l$, and H_1 which maps a string (a user's ID) to an element in G_1, that is, H_1:$\{0,1\}^* \to G_1$. Moreover, it selects $s \in Z_q^*$ as its private master key and computes the corresponding system public key as $P_{\mathrm{pub}} = sP$. Then, KGC publishes the system parameter set $\{G_1, G_2, q, \hat{e}, P, P_{\mathrm{pub}}, H, H_1, H_2\}$. After that, when a user U (sender/chooser) registers his identifier ID_U, KGC will compute a public/private key pair $U_{\mathrm{pub}}/U_{\mathrm{priv}}$ for him, where $U_{\mathrm{pub}} = H_1(ID_U)$ and $U_{\mathrm{priv}} = sU_{\mathrm{pub}}$.

(2) Oblivious Transfer Phase. In this phase, when a sender possessing n messages ($m_1, m_2, \ldots$, and m_n) wants to obliviously transfer k messages of them ($m_{\sigma_1}, m_{\sigma_2}, \ldots$, and m_{σ_k}) to a chooser, they together will execute the following steps, where the public/private key pairs of the sender and chooser are $S_{\mathrm{pub}}/S_{\mathrm{priv}}$ and $C_{\mathrm{pub}}/C_{\mathrm{priv}}$, respectively, and $\{\sigma_1, \sigma_2, \ldots, \sigma_k\} \subset \{1, 2, \ldots, n\}$ are the set of k choices selected by the chooser in advance. We also depict them in Table 1.

Step 1. The chooser randomly chooses two integers $a, b \in Z_q^*$ and computes $V = abC_{\mathrm{pub}}$, $V_j = bH(\sigma_j)C_{\mathrm{priv}}$, where $j = 1, 2, \ldots, k$ and V_j are the k random choices. After that, he generates a signature Sig on V by computing $h = H_2(V)$ and $\mathrm{Sig} = hC_{\mathrm{priv}}$. Then, he sends ID_c, V, $V_1, \ldots, V_k$ together with Sig to the sender.

Table 1: The proposed k-out-of-n authentic OT protocol.

Sender $(S_{\text{pub}}/S_{\text{priv}}(= sS_{\text{pub}}))$	Chooser $(C_{\text{pub}}/C_{\text{priv}}(= sC_{\text{pub}}))$
	(1) Selects , $b \in_R Z_q^*$, computes $V = abC_{\text{pub}}$, for $j = 1$ to k, computes $V_j = bH(\sigma_j)C_{\text{priv}}$, computes $h = H_2(V)$ and $\text{Sig} = hC_{\text{priv}}$.
$\xleftarrow{ID_C, V, V_1, \ldots, V_k, \text{Sig}}$	
(2) Computes $h = H_2(V)$ and verifies $\hat{e}(P, \text{Sig}) =? \hat{e}(P_{\text{pub}}, hC_{\text{pub}})$. If it does not hold, aborts. Selects $c \in_R Z_q^*$ and computes $U_j = cV_j$, for $j = 1, \ldots, k$, and $ct_i = m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^c$, for $i = 1, \ldots, n$.	
$\xrightarrow{U_1, \ldots, U_k, ct_1, \ldots, ct_n}$	
	(3) For $j = 1$ to k and $i = 1$ to n, computes $m_{\sigma_i} = ct_{\sigma_i} \oplus \hat{e}(U_j, S_{\text{pub}})^a$.

Step 2. After receiving IDc, V, $V_1, \ldots, V_k$ and Sig from the chooser, the sender computes $h = H_2(V)$ and verifies the chooser's signature by checking whether the equation $\hat{e}(P, \text{Sig}) = \hat{e}(P_{\text{pub}}, hC_{\text{pub}})$ holds. If it holds, he believes that the chooser is the intended party as claimed. Then, the sender randomly chooses an integer $c \in Z_q^*$ and computes $U_j = cV_j$ and $ct_i = m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^c$, where $j = 1, \ldots, k$, $i = 1, \ldots, n$, and m_i are the n messages. He/She then sends $U_1, \ldots, U_k, ct_1, \ldots,$ and ct_n to the chooser.

Step 3. After receiving the message $U_1, \ldots, U_k, ct_1, \ldots,$ and ct_n from the sender, the chooser can obtain the k intended messages by at most computing the equation, $m_{\sigma_j} = ct_{\sigma_j} \oplus \hat{e}(U_j, S_{\text{pub}})^a$, $nk - \cdot(k(k-1)/2)(= n + (n-1) + \cdots + (n - (k-1)))$ times.

(3) A Misleading Attack for Chooser's Privacy Preservation. To demonstrate the chooser's privacy more clearly, we take the following as a counterexample. According to step 1 in our protocol, the chooser computes $V_1, \ldots, V_k$, where $V_j = bH(\sigma_j)C_{\text{priv}}$ and $j = 1$ to k. Since b and C_{priv} are both the same for V_i and V_j, a misleading attack may be that $V_i/V_j = H(\sigma_i)/H(\sigma_j)$. A malicious sender can precompute $H(\sigma_i)/H(\sigma_j)$ for each i, j in the interval $[1, n]$. After receiving $V_1, \ldots, V_k$ from the chooser, he computes each V_i/V_j for all i, j in $[1, k]$ for a comparison with the precomputed values. Consequently, the sender may guess some or all of the chooser's choices. Therefore, the protocol cannot achieve chooser privacy. However, the mistake here is that both V_i and V_j are points in the additive group G_1. The division operation V_i/V_j is invalid because G_1 is an additive group.

5. Security Analysis

In this section, we use the following claims to show that our protocol not only is correct but also possesses the properties of mutual authentication, chooser's privacy, and sender's privacy and can resist against active attacks such as relay attack, man-in-the-middle attack, and denial of service attack.

Claim 1. The proposed protocol is correct.

Proof. After the protocol runs, the chooser can exactly obtain the k messages which he/she selected by computing

$$\begin{aligned} ct_{\sigma_j} \oplus \hat{e}\left(U_j, S_{\text{pub}}\right)^a &= ct_{\sigma_j} \oplus \hat{e}\left(cbH\left(\sigma_j\right)C_{\text{priv}}, S_{\text{pub}}\right)^a \\ &= ct_{\sigma_j} \oplus \hat{e}\left(H\left(\sigma_j\right)bcsC_{\text{pub}}, S_{\text{pub}}\right)^a \\ &= ct_{\sigma_j} \oplus \hat{e}\left(H\left(\sigma_j\right)abC_{\text{pub}}, sS_{\text{pub}}\right)^c \\ &= ct_{\sigma_j} \oplus \hat{e}\left(H\left(\sigma_j\right)V, S_{\text{priv}}\right)^c = m_{\sigma_j}. \end{aligned} \tag{4}$$

□

Claim 2. The proposed protocol can achieve mutual authentication.

Proof. We show the holdness of this claim by using the following two reasons.

(1) Apparently, it can be easily seen that the sender can authenticate the chooser by verifying the chooser's signature, Sig (as described in step 2 of the oblivious transfer phase).

(2) For that the ciphertext ct_i $(= m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^c)$ contains the sender's private key $S_{\text{priv}}(= sS_{\text{pub}})$, the chooser can compute the meaningful message m_{σ_j} only via using the sender's public key S_{pub} (also refer to the equation in claim 1). This means that only the true sender can produce the right ct_i and thus can be authenticated by the chooser using his public key.

□

Claim 3. The proposed protocol can achieve the chooser's privacy.

Proof. Due to the fact that each of the chooser's k choices $\sigma_j \in \{1, 2, \ldots, n\}$ are first hashed and randomized by H and b respectively, and then signed as $V_j = bH(\sigma_j)C_{\text{priv}}$ by chooser C in step 1, where b is a random number. We argue that nobody except for the chooser can know the choice σ_j. Because even an attacker might steal the chooser's private key C_{priv}, he/she cannot obtain $bH(\sigma_j)$ from V_j owing to the hardness of ECDLP. That is, he cannot figure out $bH(\sigma_j)$, and therefore not to mention σ_j. More formally, let $\mathcal{A} = \{(b, \sigma_j) \in Z_q * Z_n \mid bH(\sigma_j)C_{\text{priv}} = V_j\}$; that is, $\mathcal{A}$ consists of all the possible ordered pairs (b, σ_j) satisfying the equation $bH(\sigma_j)C_{\text{priv}} = V_j$. If we are given a value V_j, then under fixed C_{priv}, there only exists a unique value $bH(\sigma_j)$ satisfying the equation. And for a given $bH(\sigma_j)$, under the definition of a collision-free one-way hash function, once σ_j has been determined, the value of b is determined as well. That is, the relationship between b and σ_j is one-to-one. Having this observation in mind and the dimension of σ_j is n, we can see that there are n (b, σ_j) pairs in $\mathcal{A}$. In other words, $\Pr[\sigma_j \mid V_j] = \Pr[\sigma_j] = 1/n$ which means that, under seeing a specific V_j, the choice σ_j of the chooser cannot be revealed other than guessing. This achieves the *Shannon perfect secrecy*. Therefore, the proposed protocol possesses chooser's privacy. □

Claim 4. The proposed scheme can achieve the sender's privacy.

Proof. Assume that malicious chooser $\hat{C}$ wants to obtain more than k messages in the protocol. If he/she could succeed, then, the sender's privacy is violated (see Section 2.1). However, we will prove that, other than his k chosen messages, it is computationally infeasible for $\hat{C}$ to obtain the $(k+1)$th message by using the following two arguments, (I) and (II). In argument (I), we show why $\hat{C}$ must follow the protocol to form the values of V and $kV_j s$; otherwise, he/she cannot obtain the k chosen messages. In argument (II), we show that if $\hat{C}$ intends to obtain the $(k+1)$th message, he/she will face the intractable CTCDH problem under the assumption that $H(\cdot)$ is a random hash function. □

Argument (I). $\hat{C}$ must follow the protocol to form the values of $V(= ab\hat{C}_{\text{pub}})$ and $V_j(= bH(\sigma_j)\hat{C}_{\text{priv}})$, for $j = 1$ to k; otherwise, he cannot obtain the k chosen messages, $m_{\sigma_1}, \ldots, m_{\sigma_j}$.

In the following, we further divide this argument into three cases: (a) $\hat{C}$ fakes V but forms V_j honestly, (b) $\hat{C}$ fakes V_j but forms V honestly, and (c) $\hat{C}$ fakes both the values of V and V_j. (For each case's explanation, refer to Table 1.)

(a) $\hat{C}$ fakes V but forms V_j honestly. Assume that $\hat{C}$ is dishonest in forming V but forms V_j honestly as specified in the protocol. For example, without loss of generality, it replaces V with a specific $X \in G_1$ and computes $V_j = bH(\sigma_j)\hat{C}_{\text{priv}}$. Then, the sender will compute $U_j = cV_j$, $ct_i = m_i \oplus \hat{e}(H(i)X, S_{\text{priv}})^c$ and send them back to $\hat{C}$. As a result, $\hat{C}$ cannot decrypt $ct_{\sigma_j}(ct_{\sigma_j} = m_{\sigma_j} \oplus \hat{e}(U_j, S_{\text{pub}})^a)$ to obtain the k messages since $\hat{e}(U_j, S_{\text{pub}})^a$ is obviously not equal to $\hat{e}(H(\sigma_j)X, S_{\text{priv}})^c$ (refer to claim 1). Perhaps, for obtaining the k messages, $\hat{C}$ may try another way by computing $\hat{e}(H(i)X, S_{\text{priv}})^c$ expected to be equal to $\hat{e}(U_j, S_{\text{pub}})^a$. But this is computationally infeasible since $\hat{C}$ does not know both the sender's private key S_{priv} and the one-time secrecy c. To extract c from U_j is an ECDLP.

(b) $\hat{C}$ fakes $V_j s$ but forms V honestly. Assume that $\hat{C}$ is dishonest in forming $V_j s$ but forms V in the same manner as specified in the protocol. For example, without loss of generality, he replaces each V_j with a specified $X_j \in G_1$ and computes $V = ab\hat{C}_{\text{pub}}$. Then, the sender will compute $U_j = cV_j = cX_j$, $ct_i = m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^c = m_i \oplus \hat{e}(H(i)ab\hat{C}_{\text{pub}}, S_{\text{priv}})^c$, for $i = 1$ to n, and send them back to $\hat{C}$. As a result, $\hat{C}$ cannot decrypt ct_{σ_j} since $\hat{e}(U_j, S_{\text{pub}})^a = \hat{e}(cX_j, S_{\text{pub}})^a$ is obviously not equal to $\hat{e}(H(i)V, S_{\text{priv}})^c$. Perhaps, for obtaining the k messages, $\hat{C}$ may try another way by computing $\hat{e}(H(i)V, S_{\text{priv}})^c$ $(= \hat{e}(H(i)ab\hat{C}_{\text{pub}}, S_{\text{priv}})^c)$ expected to be equal to $(U_j, S_{\text{pub}})^a$. But again this is computationally infeasible since $\hat{C}$ does not know both the sender's private key S_{priv} and the one-time secrecy c. Even he knows S_{priv}, extracting c from $U_j(= cX_j)$ is an ECDLP. Hence, $\hat{C}$ cannot compute the value $\hat{e}(H(i)V, S_{\text{priv}})^c$ to decrypt ct_{σ_j} for obtaining the k messages, m_{σ_j}.

(c) $\hat{C}$ fakes both the values of V and V_j. Without loss of generality, we assume that $\hat{C}$ replaces V with X and also fakes V_j as $H(\sigma_j)X$. Under this construction, the value of U_j computed by the sender would be $U_j = cV_j = cH(\sigma_j)X$ and the ciphertexts ct_{σ_j} would be $m_{\sigma_j} \oplus \hat{e}(H(\sigma_j)X, S_{\text{priv}})^c$, for $j = 1$ to k, or equivalently, $ct_{\sigma_j} = m_{\sigma_j} \oplus \hat{e}(cH(\sigma_j)X, S_{\text{priv}})$. Although, $\hat{C}$ knows the value of $cH(\sigma_j)X$ (since it just equals to U_j received from the sender), it still cannot compute $\hat{e}(cH(\sigma_j)X, S_{\text{priv}})$ without the knowledge of S_{priv}. From above description, we know that when the setting of V is X and V_j is $H(\sigma_j)X$, $\hat{C}$ cannot obtain m_{σ_j}. Not to mention, $\hat{C}$ might set V_j as $H(\sigma_j)Y$, where $Y(\neq X)$ is a random chosen element in G_1. In summary, $\hat{C}$ cannot obtain the k selected messages under the violation of setting both the values, V and V_j.

Argument (II). If $\hat{C}$ follows the protocol honestly to obtain k messages, but intends to extract the $(k+1)$th message then it will face the intractable CTCDH problem under the assumption that $H(\cdot)$ is a random hash function.

That $\hat{C}$ wants to obtain message m_i implies $\hat{C}$ would have the knowledge of $\hat{e}(H(i)V,\ S_{\text{priv}})^c$ $(= \hat{e}(U_j,\ S_{\text{pub}})^a)$ (in fact, according to argument (I), an honest chooser C could know k of the n values, $\hat{e}(H(i)V,\ S_{\text{priv}})^c$, for $i = 1$ to n, since $\hat{e}(H(i)V,\ S_{\text{priv}})^c = \hat{e}(U_j,\ S_{\text{pub}})^a$, for $i = \sigma_j$ and $j = 1$ to k). Let $y^{(i)} \in G_2$ and $\hat{e}(H(i)V,\ S_{\text{priv}})^c = y^{(i)}$. According to argument (I), for obtaining the k chosen messages, $\hat{C}$ cannot change the structures of $V(= abC_{\text{pub}})$ and V_j $(= bH(\sigma_j)\hat{C}_{\text{priv}})$. Under this situation, $y^{(i)}$ only can be decomposed as $y^{(i)} = \hat{e}(H(i)ab\hat{C}_{\text{pub}},\ S_{\text{priv}})^c = \hat{e}(abH(i)\hat{C}_{\text{priv}},\ S_{\text{pub}})^c$ since $S_{\text{priv}} = sS_{\text{pub}}$ and $\hat{C}_{\text{priv}} = s\hat{C}_{\text{pub}}$. Moreover, under the assumption that $H(\cdot)$ is a random hash function and the fact that $\hat{C}$ has the knowledge of a, b, $\hat{C}_{\text{priv}}$, and S_{pub}, $y^{(i)}$ can be represented as $(g_i)^c$, where g_i equals to $\hat{e}(abH(i)\hat{C}_{\text{priv}},\ S_{\text{pub}})$ and is a random element in G_2 due to the assumption that $H(\cdot)$ is a random hash function. Consequently, the problem $\hat{C}$ really faces is finding the $(k+1)$th pair $(\sigma_{k+1},\ (g_{\sigma_{k+1}})^c)$ with the knowledge of k pairs of $(\sigma_1,\ (g_{\sigma_1})^c), (\sigma_2,\ (g_{\sigma_2})^c), \ldots,$ and $(\sigma_k,\ (g_{\sigma_k})^c)$, where $(g_{\sigma_j})^c = \hat{e}(U_j,\ S_{\text{pub}})^a$, but without the knowledge of sender's one-time secrecy c (since it is an ECDLP for extracting c from $U_j(= cV_j)$). This is known as the intractable CTCDH problem introduced in Section 2.3 by letting $k = (l-1)$. Therefore, the chooser cannot obtain the $(k+1)$th message.

According to arguments I and II, we have proven claim 4 that our scheme has the sender's privacy.

Claim 5. The proposed scheme can resist against replay attack.

Proof. Suppose that an adversary intercepts a chooser's OT request (containing ID_C, V, V_j, and Sig) and replays it later. After receiving the sender's new response $(U_1, \ldots, U_k, ct_1, \ldots, ct_n)$ computed from the replayed V and V_j, the adversary cannot obtain the k selected messages by computing $m_{\sigma_j} = ct_{\sigma_j} \oplus \hat{e}(U_j,\ S_{\text{pub}})^a$ since he/she does not know the value of a embedded in the replayed message V. It is computationally infeasible for the adversary to extract a from $V = abC_{\text{pub}}$, due to the hardness of ECDLP. □

Claim 6. The proposed scheme can resist against man-in-the-middle attack (MIMA).

Proof. MIMA is an attack that an adversary E intercepts the communication line between two communicating parties and uses some means to make them believe that they each are talking to the intended party as claimed. But indeed, they are talking to E. Figure 1 illustrates the scenario of such a MIMA. We first argue that the adversary E cannot succeed in this scenario since it cannot generate the valid message (2), $(\text{ID}_C,\ V', V'_1, \ldots,\ V'_k, \text{Sig}')$ as shown in the figure. More clearly, without the knowledge of chooser's private key C_{priv}, he/she cannot forge a valid signature Sig′ in message (2) to be successfully verified by the sender since Sig′ should be equal to $H_2(V)\ C_{\text{priv}}$. In addition, it is also hard for E to forge valid message (4), $(U'_1, \ldots,\ U'_k,\ ct'_1, \ldots,\ ct'_n)$, to be accepted by the chooser. Since that for embedding a meaningful m'_i into ct'_i, E must have the knowledge of $\hat{e}(H(i)V,\ S_{\text{priv}})^c$. Although E can choose another random nonce c' such that $U'_j = c'V_j$, it still has to know the sender's private key S_{priv} to form the valid $ct'_i (= m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^{c'})$. Therefore, without the knowledge of S_{priv}, E cannot launch such a MIMA attack. □

Claim 7. The proposed scheme can resist a denial of service attack (DOS).

Proof. Our protocol has a built-in mutual authentication property; thus, it can prevent this kind of attack, as the sender needs only one hash and two bilinear pairing computations to authenticate the chooser in step (2). Once the sender finds that the authenticating equation $\hat{e}(P, \text{Sig}) = \hat{e}(P_{\text{pub}},\ hC_{\text{pub}})$ does not hold, it aborts the procedure. □

5.1. Communicational Cost Comparisons. Generally, the communicational cost of a protocol run consists of three factors: (1) needed passes, (2) computational overhead, and (3) needed transmission data size (NTDS) or bandwidth consumption. It is well known that factor (1) is always dominant over factor (2). Hence, in this section, we focus only on factor (1) and (3) to demonstrate the communication cost comparisons among our nonadaptive OT_k^n protocol and the other same type OT_k^n protocols, such as Chu and Tzeng's [16] (which is to our best knowledge, the most efficient OT_k^n scheme up to date), Mu et al.'s [13], Naor and Pinkas's [7], and recent works [17, 18, 24, 27]. From factor (1), our scheme is the most efficient since it only requires two passes. As to factor (3), the data size transmitted in our scheme is also the minimal among such type of OT_k^n schemes. For demonstrating this in the following, we will first describe two underlying facts and used notations for making comparisons about factor (3).

Generally speaking, we have the following two facts for cryptosystems.

Fact 1. To the same security level, a RSA cryptosystem would require a key length of 1024 bits while an ECC-based cryptosystem only needs 160 bits.

Fact 2. The length of the ciphertexts for RSA, ElGamal, and ECC-based cryptosystems is 1024 bits, 1024 bits, and 160 bits, correspondingly.

Notations. We use |string/action| to represent the bit length of a *string*, or the required bit length that an *action* performs.

After the description of used facts and notations, we now use them to estimate the needed transmission data size (NTDS) of our scheme and the above-mentioned OT_k^n protocols. In our scheme, each of the variables V, $V_1, \ldots, V_k$, Sig, $U_1, \ldots, U_k$ transmitted between the chooser and sender is an ECC point. Thus, the NTDS from the chooser to the sender is estimated as $160 * (k+2)$ bits and from the sender to the chooser is $160\,k + n * |\text{ciphertext}|$ bits. Naor and Pinkas's scheme [7] constructs their OT_k^n scheme by evoking an OT_1^2 primitive $\log n$ times. Thus, the needed number of passes is $\log n$ times the number of passes required in one of their OT_1^2's protocol run and

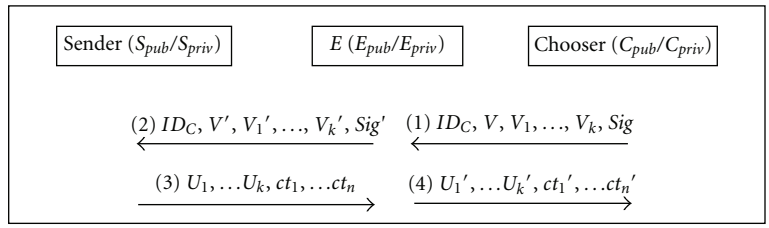

Figure 1: The scenario of MIMA attack.

Table 2: Needed rounds and data size comparisons among OT_k^n protocols.

Protocol	Passes	Size of message: $C \rightarrow S$ (bits)	Size of message: $S \rightarrow C$ (bits)	Mutual authentication
Ours	2	$160 * (k+2)$	$160k + n * \lvert\text{ciphertext}\rvert$	Yes
Naor and Pinkas [7]	$k* \log n$ OT_1^2	depends on OT_1^2	depends on OT_1^2	No
Mu et al.'s scheme (1) [13]	3	$1024k$	$1024n + nk * \lvert\text{ciphertext}\rvert$	No
Mu et al.'s scheme (2) [13]	2	$1024 * 2n$	$n * \lvert\text{ciphertext}\rvert$	No
Chu and Tzeng [16]	2	$1024k$	$1024 * (k + 1) + n * \lvert\text{ciphertext}\rvert$	No
Zhang and Wang [17]	2	$1024 * (k+3)$	$1024n + n * \lvert\text{ciphertext}\rvert$	No
Huang and Chang [18]	3	$1024k$	$(n + k) * \lvert\text{ciphertext}\rvert$	
Camenisch et al. [24]	$2 + k * \text{Pok}$	$\lvert\text{Pok}\rvert + k * \lvert\text{BlindExtract}\rvert$	$n * \lvert\text{ciphertext}\rvert + \lvert\text{Pok}\rvert + k * \lvert\text{BlindExtract}\rvert$	No
Chang and Lee [27]	4	$1024k$	$(n + 2k + 2)* 1024$	No
Ma et al. [28]	$k * (2 + \text{Pok})$	$k * 3\lvert\text{ciphertext}\rvert$	$k * (n * 2\lvert\text{ciphertext}\rvert)$	No

likewise the NTDS is about $\log n$ times of the NTDS that an OT_1^2's work demands. Therefore, their scheme has the most expensive communicational cost. As for Camenisch et al.'s protocol [24], the communicational cost is expensive as well due to the complexity of the protocol. In their protocol, the sender first sends n commitments to the chooser, and then the sender and the chooser together run a proof-of-knowledge (Pok) subprotocol for assuring the correctness of the commitments. If the proof is valid, the sender sends n ciphertexts to the chooser, and the chooser then runs the BlindExtract subprotocol k times with the help of the sender to extract the blind choices to decrypt the ciphertexts.

Consequently, the number of passes for executing protocol [24] is $2 + k * \text{Pok}$, where Pok represents the required passes for executing the proof-of-knowledge subprotocol. Besides, the NTDS from chooser to sender is estimated as $\lvert\text{Pok}\rvert + k * \lvert\text{BlindExtract}\rvert$ and from sender to chooser is $n * \lvert\text{ciphertext}\rvert + \lvert\text{Pok}\rvert + k * \lvert\text{BlindExtract}\rvert$. Similarly, the passes and NTDS of other studies can be estimated in the same manner. We show the comparison results in Table 2.

From Table 2, we can see that our scheme not only possesses the mutual authentication function but also is the most efficient in both needed passes and NTDS among these related. Therefore, our scheme can be gracefully used when applied in commercial applications (e.g., Kerschbaum et al.'s method [1] used OT scheme as a building block in constructing RFID benchmarking protocols).

6. Conclusion

An OT scheme which is secure and efficient in communicational cost is essential and eager for commercial applications. After reviewing most of the OT schemes, we found that, other than considering the protocol's correctness and privacy of both communication parties, almost all of them lack the security services, such as mutual authentication, and the prevention of replay, DOS, and main-in-the-middle attacks. Hence, they should run under a secure channel when applied in commercial applications. This will increase execution overhead. Therefore, to get rid of using the secure channel (for improving the communicational efficiency in some applications, such as mental poker playing, oblivious key searching), we propose a novel k-out-of-n oblivious transfer protocol by combining an OT scheme with a security mechanism based on bilinear pairing. We have proved that our scheme not only is correct but also possesses the properties of mutual authentication, the sender's privacy, and the chooser's privacy and can resist against replay and MIMA attacks. Further, we have compared our scheme with other nonadaptive k-out-of-n OT schemes in the aspects of needed passes, NTDS, and the function of mutual authentication and shown the result in Table 2. From Table 2, we can see that our scheme is the most efficient in communicational cost (including needed passes and NTDS). In addition, to our knowledge, it is the only OT_k^n scheme that has successfully integrated the function of mutual authentication nowadays.

References

[1] F. Kerschbaum, N. Oertel, and L. W. F. Chaves, "Privacy-preserving computation of benchmarks on item-level data using RFID," in *Proceedings of the 3rd ACM Conference on Wireless Network Security (WiSec '10)*, pp. 105–110, March 2010.

[2] M. O. Rabin, "How to exchange secrets with oblivious transfer," Tech. Rep. TR-81, Aiken Computation Lab, Harvard University, Cambridge, Mass, USA, 1981.

[3] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM*, vol. 28, no. 6, pp. 637–647, 1985.

[4] G. Brassard, C. Crepeau, and J.-M. Robert, "All-or-nothing disclosure of secrets," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '86)*, vol. 263 of *Lecture Notes in Computer Science*, pp. 234–238, 1986.

[5] J. S. Chou and Y. S. Yeh, "Mental poker game based on a bit commitment scheme through network," *Computer Networks*, vol. 38, no. 2, pp. 247–255, 2002.

[6] M. Bellare and S. Micali, "Non-interactive oblivious transfer and application," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '89)*, vol. 435 of *Lecture Notes in Computer Science*, pp. 547–557, 1989.

[7] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '99)*, Lecture Notes in Computer Science, pp. 573–590, 1999.

[8] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," in *Proceedings of the 1st ACM Conference on Electronic Commerce*, 1999.

[9] M. Naor and B. Pinkas, "Distributed oblivious transfer," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '00)*, vol. 1976 of *Lecture Notes in Computer Science*, 2000.

[10] M. Naor and B. Pinkast, "Oblivious transfer and polynomial evaluation," in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (FCRC '99)*, pp. 245–254, May 1999.

[11] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the 12th annual ACM-SIAM symposium on Discret Mathematics (SODA '01)*, pp. 448–457, 2001.

[12] H. Ghodosi, "On insecurity of Naor-Pinkas' distributed oblivious transfer," *Information Processing Letters*, vol. 104, no. 5, pp. 179–182, 2007.

[13] Y. Mu, J. Zhang, and V. Varadharajan, "m out of n oblivious transfer," in *Proceedings of the 7th Australasian Conference on Information Security and Privacy (ACISP '02)*, vol. 2384 of *Lecture Notes in Computer Science*, pp. 395–405, 2002.

[14] H. Ghodosi and R. Zaare-Nahandi, "Comments on the 'm out of n oblivious transfer'," *Information Processing Letters*, vol. 97, no. 4, pp. 153–155, 2006.

[15] W. Ogata and K. Kurosawa, "Oblivious keyword search," *Journal of Complexity*, vol. 20, no. 2-3, pp. 356–371, 2004.

[16] C. K. Chu and W. G. Tzeng, "Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries," in *Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC '05)*, pp. 172–183, January 2005.

[17] J. Zhang and Y. Wang, "Two provably secure k-out-of-n oblivious transfer schemes," *Applied Mathematics and Computation*, vol. 169, no. 2, pp. 1211–1220, 2005.

[18] H. F. Huang and C. C. Chang, "A new design for efficient t-out-n oblivious transfer scheme," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA '05)*, pp. 28–30, March 2005.

[19] A. Parakh, "Oblivious transfer using elliptic curves," in *Proceedings of the 15th International Conference on Computing (CIC '06)*, pp. 323–328, November 2006.

[20] S. Kim and G. Lee, "Secure verifiable non-interactive oblivious transfer protocol using RSA and Bit commitment on distributed environment," *Future Generation Computer Systems*, vol. 25, no. 3, pp. 352–357, 2009.

[21] Y. F. Chang and W. C. Shiao, "The essential design principles of verifiable non-interactive OT protocols," in *Proceedings of the 8th International Conference on Intelligent Systems Design and Applications (ISDA '08)*, pp. 241–245, November 2008.

[22] L. M. Kohnfelder, "On the signature reblocking problem in public-key cryptography," *Communications of the ACM*, vol. 21, no. 2, p. 179, 1978.

[23] S. Halevi and Y. T. Kalai, "Smooth projective hashing and two-message oblivious transfer," Cryptology ePrint Archive 2007/118, 2007.

[24] J. Camenisch, G. Neven, and A. Shelat, "Simulatable adaptive oblivious transfer," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 4515 of *Lecture Notes in Computer Science*, pp. 573–590, 2007.

[25] M. Green and S. Hohenberger, "Blind identity-based encryption and simulatable oblivious transfer," Cryptology ePrint Archive 2007/235, 2007.

[26] J. Qin, H. W. Zhao, and M. Q. Wang, "Non-interactive oblivious transfer protocols," in *Proceedings of the International Forum on Information Technology and Applications (IFITA '09)*, pp. 120–124, May 2009.

[27] C. C. Chang and J. S. Lee, "Robust t-out-of-n oblivious transfer mechanism based on CRT," *Journal of Network and Computer Applications*, vol. 32, no. 1, pp. 226–235, 2009.

[28] X. Ma, L. Xu, and F. Zhang, "Oblivious transfer with timed-release receiver's privacy," *Journal of Systems and Software*, vol. 84, no. 3, pp. 460–464, 2011.

[29] W. Stallings, *Cryptography and Network Security—Principals and Practices*, Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2003.

[30] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the 40th annual ACM symposium on Theory of Computing (STOC '82)*, pp. 365–377, 1982.

[31] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '01)*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, 2001.

[32] D. R. Stinson, *Cryptography—Theory & Practice*, Chapman & Hall/CRC Taylor & Francis Group, 3rd edition, 2006.

[33] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-group signature scheme," in *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography*, vol. 2567 of *Lecture Notes in Computer Science*, pp. 31–46, 2003.

[34] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, "The one-more-RSA-inversion problems and the security of chaum's blind signature scheme," in *Proceedings of Financial Cryptography (FC '01)*, vol. 2248 of *Lecture Notes in Computer Science*, pp. 319–338, 2003.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2012, Article ID 296234, 12 pages
doi:10.1155/2012/296234

Research Article

Parlay X Web Services for Policy and Charging Control in Multimedia Networks

Ivaylo Atanasov, Evelina Pencheva, and Dora Marinska

Department of Telecommunications, Technical University of Sofia, 1000 Sofia, Bulgaria

Correspondence should be addressed to Evelina Pencheva, enp@tu-sofia.bg

Received 4 November 2011; Accepted 27 March 2012

Academic Editor: Mohamed Hamdi

The paper investigates the capabilities of Parlay X Web Services for Policy and Charging Control (PCC) in managing all Internet-protocol-based multimedia networks (IMSs). PCC is one of the core features of evolved packet networks. It comprises flow-based charging including charging control and online credit control, gating control, and Quality of Service (QoS) control. Based on the analysis of requirements for PCC, the functionality for open access to QoS management and advanced charging is identified. Parlay X Web Services are evaluated for the support of PCC, and some enhancements are suggested. Implementation aspects are discussed, and Parlay X interfaces are mapped onto IMS control protocols. Use cases of Parlay X Web Services for PCC are presented.

1. Introduction

IMS stands for internet protocol multimedia subsystem which is an architectural framework for service delivery in evolved packet networks. IMS enables various types of multimedia services based on access independency and IP connectivity [1]. The main requirement for IMS in conjunction with IP connectivity access network (IP-CAN) is to provide quality of service. Quality of service (QoS) is used to differentiate multimedia offering from traditional Internet services, which in most cases do not provide QoS. In order to provide a mechanism for service-aware QoS control and coherent charging, the Policy and Charging Control architecture is standardized. The Policy and Charging Control (PCC) is a key concept in IMS architecture and it is designed to enable flow-based charging, including, for example, online credit control, as well as policy control, which includes support for service authorization and QoS management [2].

In IMS, the user equipment negotiates with the network the session parameters by means of Session Initiation Protocol (SIP) signaling [3]. The service-related information is delivered to PCC functional entities and is used to form authorized IP QoS data (e.g., maximum bandwidth and QoS class) and charging rules as well as user plane event reporting (e.g., bearer loss recovery, access network change, and out of credit) for any access network [4].

To stimulate service provisioning and to allow applications outside of the network operator domain to invoke communication functions, an approach to opening the network interfaces is developed [5]. The open access to network functions allows 3rd party applications to make use of network functionality and to receive information from the network through application programming interfaces (APIs). Parlay X Web Services are highly abstracted means for access to network functionality [6]. Parlay X provides APIs for a palette of network functions such as call control, data session control, mobility, messaging, QoS control, and charging.

In this paper, we assess the support of existing Parlay X Web Services for access to PCC functions in multimedia networks.

The paper is structured as follows. Some related works are discussed in Section 2. The PCC architecture with User Data Convergence is discussed in Section 3. Based on the PCC architectural framework, the requirements for

open access to flow-based charging and policy control are summarized in Section 4. The standardized capabilities of Parlay X Web Services for open access to PCC are evaluated in Section 5. In Section 6, some enhancements to Parlay X Web Services are suggested having in mind the identified requirements. The Parlay X interfaces implementation requires mapping of interfaces methods onto network control protocols messages. Such mapping does not exist as the PCC specifications are defined after the specification of Parlay X Web Services. The suggested mapping is sketched in Section 7. The Parlay X interfaces applicability is illustrated by typical use cases. Finally, Section 8 concludes by highlighting the benefits of third party QoS management in IP-based multimedia networks.

2. Related Work

PCC allows flexible QoS management of ongoing multimedia sessions in case of changing both the access networks and user devices with different capabilities. The PCC can also contribute to seamless service continuity in case of handover between two wireless networks without user intervention and with minimal service disruptions.

Good and Ventura [7] propose a multilayered policy control architecture that extends the general resource management function being standardized; this extended architecture gives application developers greater control over the way the services are treated in the transport layer. Good et al. [8] suggest enhancements to the PCC framework that extend the end-to-end inter-domain mechanisms to discover the signaling routes at the service control layer and use this to determine the paths traversed by the media at the resource control layer. Because the approach operates at these layers, it is compatible with existing transport networks and exploits already existing QoS control mechanisms. In [9], it is presented an architecture with policy-based network management focusing on access network optimization while taking service level agreements (SLAs), business objectives, routing rules, service information, user profiles, and platform conditions into account. Zhao et al. [10] present a policy-based radio resources allocation scheme. Different channel allocation algorithms and channel allocation strategies form a series of policies, thus constituting a policy-based channel allocation scheme. A policy-based service provisioning system is proposed [11] in order to provide different classes of services.

The necessity of open access to QoS control is substantiated in [12]. Stojanovic et al. [13] address an open issue of end-to-end service specification and mapping in next-generation networks. A centralized approach has been considered via the third party agent that manages the negotiation process in a group of domains. The authors suggest a general structure of the service specification form, which contains technical parameters related to a particular service request. Bormann et al. [14] extend the mediation layer between the operators core network and the charging system by adding capabilities for online charging control. The authors present a prototype that implements and extends parts of the standardized PCC architecture by the use of the open source JAIN SLEE-based framework Mobicents. Akhatar [15] develops a system and method for providing QoS enablers for 3rd party applications. In one embodiment, the method comprises user equipment establishing a session with a third party application server hosting a selected third party application and receiving from the third party application server QoS information comprising at least one of the pluralities of QoS attributes and configuring a QoS of a radio access network in accordance with the obtained QoS information. The method further comprises activating the radio access network QoS for the selected application and establishing an application session with the third party application server via the radio access network. Koutsopoulou et al. [16] present a platform that extends the existing charging collection information mechanisms and billing systems to provide for advanced and flexible charging mechanisms and pricing policies. An approach to per-flow charging with increased scalability of QoS support charging is suggested in [17].

The Parlay X "Application-Driven Quality of Service" (ADQ) [18], defined in 3GPP TS 29.199-17, allows applications to control the QoS available on user connection. It may be used for dynamic management of QoS parameters available on multimedia sessions.

The Parlay X "Payment" Web Service [19], defined in 3GPP TS 29.199-6, supports payment reservation, prepaid payments, and postpaid payments. It may be used for charging of both volume and currency amounts, a conversion function and a settlement function in case of a financially resolved dispute.

The Parlay X interfaces are defined before the standardization of IMS PCC. The analysis of PCC functions shows that these interfaces do not cover all QoS management functions that network operator can expose.

3. Architecture for Open Access to Policy and Charging Control

A possible deployment of Parlay X Web Services in PCC architecture is shown in Figure 1.

Policy and Charging Control architecture is defined in 3GPP TS 23.203 specifications [20]. The Policy and Charging Rule Function (PCRF) encompasses policy control decision and flow-based charging control functionalities. The Policy and Charging Enforcement Function (PCEF) includes service data flow detection, policy enforcement, and flow-based charging functions. It is located at the media gateway. The Online Charging System (OCS) performs online credit control functions. It is responsible for interacting in real time with the user's account and for controlling or monitoring the charges related to service usage. Offline Charging System (OFCS) is responsible for charging process where charging information is mainly collected after the end of the session and it does not affect in real time the service being used.

The Home Subscriber Server (HSS) contains all subscription-related information needed for PCC rules. If the PCC architecture supports User Data Convergence

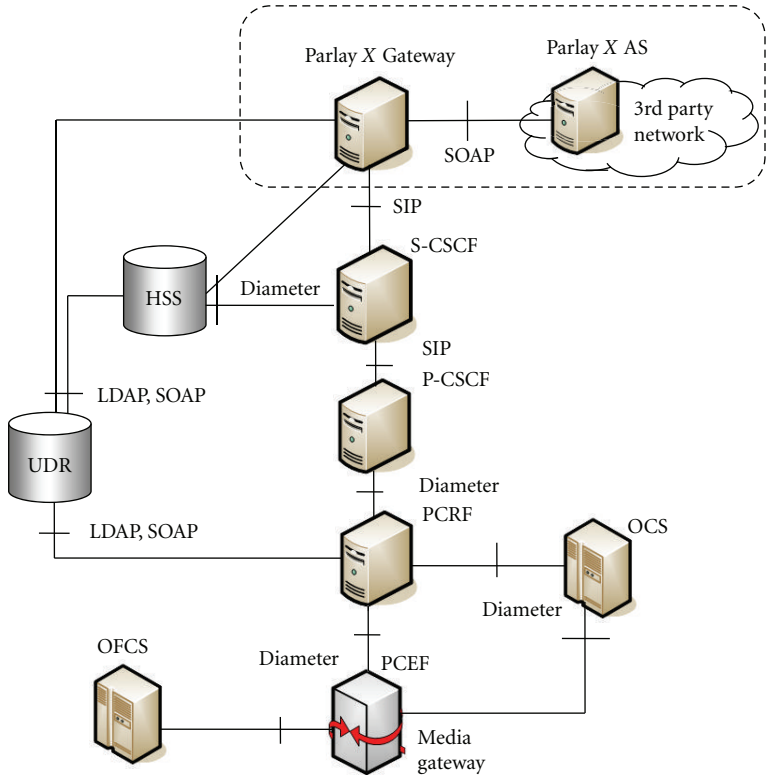

Figure 1: Deployment of Parlay X Web Services in PCC.

(UDC) defined in 3GPP TS 23.335 [21], then the User Data Repository (UDR) acts as a single logical repository for user data. The user data may, for example, contain information about default QoS parameters that have to be applied each time the user creates a session. Functional entities such as HSS and Application Servers keep their application logic, but they do not locally store user data permanently.

Call Session Control Functions (CSCFs) include functions that are common for all services. The Proxy CSCF (P-CSCF) is the first point of contact for user equipment. It deals with SIP compression, secured routing of SIP messages, and SIP sessions monitoring. Serving CSCF (S-CSCF) is responsible for user registration and session management.

Application Servers (ASs) run 3rd party applications that are outside the network operator domain. Parlay X Gateway is a special type of AS that provides Web Services interfaces for 3rd party applications and supports IMS protocols toward the network.

Diameter [22] is the control protocol in interfaces where authentication, authorization, and accounting functions are required. The control protocol in interfaces where session management is performed is Session Initiation Protocol (SIP) [23]. Lightweight Data Access Protocol (LDAP) and Simple Object Access Protocol (SOAP) are the control protocols used to create, read, modify, and delete user data in the UDR and to subscribe for and receive notifications about user data changes [24].

Note that not all charging-related interfaces and policy control functions are shown in Figure 1 for the sake of simplicity.

In Section 4, we study the functionalities of PCC and UDC in order to determine the requirements for open access to QoS management.

4. Requirements for Open Access to Policy and Charging Control

The PCC includes mechanisms for controlling the bearer traffic by using IP policies.

4.1. Gating and QoS Control. During the multimedia session establishment and modification, the user equipment negotiates a set of media characteristics. If the network operator applies policy control, then the P-CSCF sends the relevant session description information to the PCRF in order to form IP QoS authorization data. The 3rd party application can be involved in the process of QoS authorization by requesting specific QoS parameters to be applied, modified, or removed. Figure 2 illustrates the application control on QoS resource authorization for given SIP session.

Functional Requirement 1. During the SIP session establishment, 3rd party application may require to apply or to modify temporary specific QoS features on user session(s).

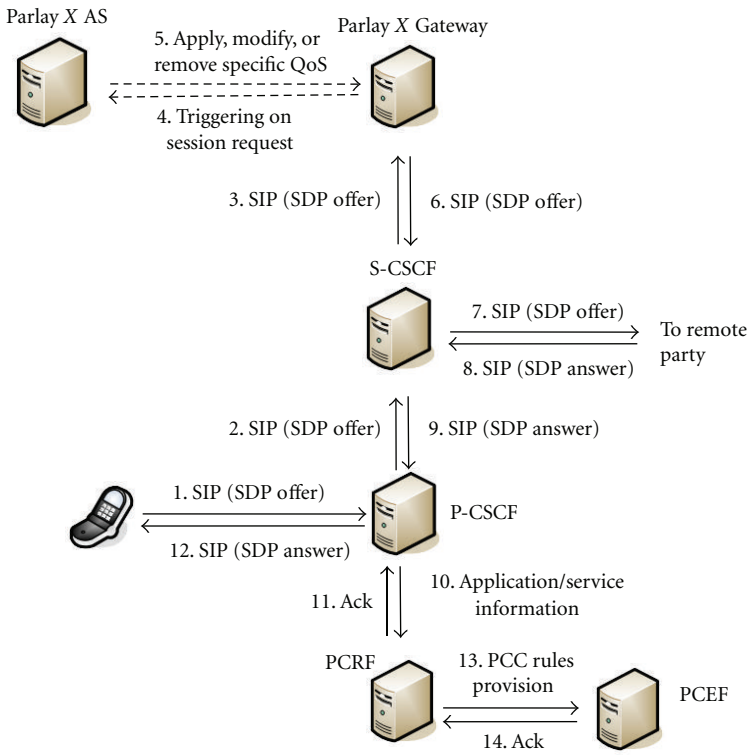

Figure 2: Application control on QoS during session establishment.

The required functions include applying temporary QoS parameters, modifying temporary QoS parameters, and removing QoS parameters for a predefined duration (e.g., for session duration). The application logic is activated in case of session initiation, modification, or termination.

In IMS, it is primary the network that decides what kind of bearer the user equipment needs during communication. Having application/service information and based on subscription information and policies, PCRF provides its decision in a form of PCC rules, which are used by the PCEF for gating control.

Any QoS events, such as indication of bearer release or bearer loss/recovery, are reported by the PCEF to the PCRF and P-CSCF. Using the policy control capabilities, the P-CSCF is able to track status of the IMS signaling and user plane bearers that the user equipment currently uses and to receive notifications when some or all service data flows are deactivated. To receive notifications about QoS events the 3rd party application needs to manage its subscriptions for notifications. By using information about bearers and signaling path status, the 3rd party application can improve service execution.

For example, the application can initiate session release on behalf of the user after indication that all service flows assigned to the ongoing session are released, but the P-CSCF has not received session termination request from the user itself. The scenario is shown in Figure 3.

Functional Requirement 2. The required functions for 3rd party application to manage the QoS event subscription include the following: creating notifications and setting the criteria for QoS; changing notifications by modification of the QoS event criteria; enabling/disabling notifications, and querying for the event criteria set; reporting notifications upon QoS event occurrences.

Functional Requirement 3. The 3rd party application should be able to request QoS resource release. Using this function, the application can prevent unauthorized bearer resources after SIP session termination.

4.2. Usage Monitoring. The 3rd party application may be interested in the accumulated usage of network resources on per-IP-CAN-session and user basis. This capability may be required for applying QoS control based on the total network usage in real time. For example, the 3rd party application may change the charging rate based on the resource usage (e.g., applying discounts after a specified volume have been reached). Another example is the assignment of a common quota for both fixed and mobile accesses for a limited time period for a defined set of subscriptions. During each session the network elements monitor the common quota, which may be consumed by one or more devices over either the wireless or fixed networks. When a defined percentage of the common quota and/or all common quota has been consumed, the 3rd party application may be notified of the event. When the common quota has been consumed the 3rd party application may block the access to the services.

Functional Requirement 4. The 3rd party application should be able to set the applicable thresholds for monitoring. Usage monitoring, if activated, will be performed for a

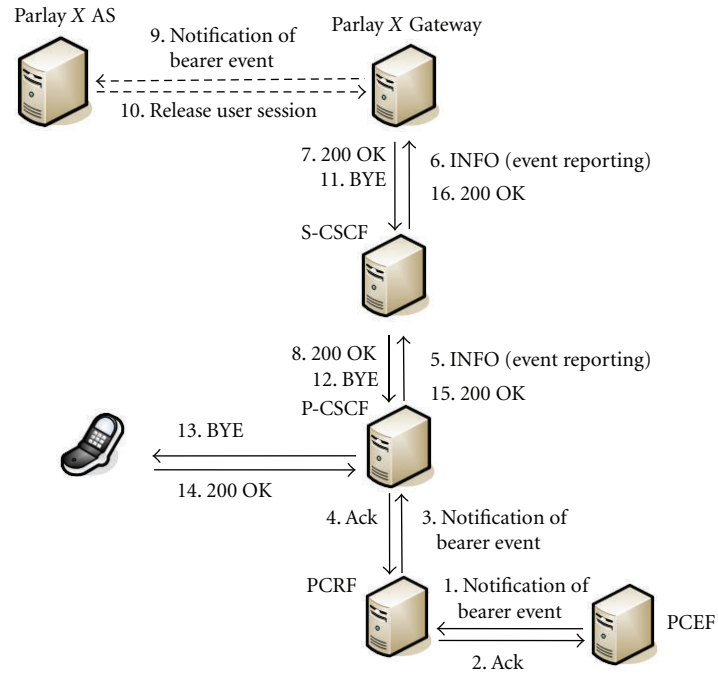

Figure 3: Notification of QoS resource release and application-initiated session release.

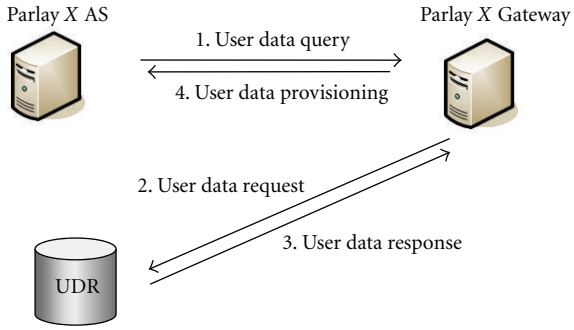

Figure 4: Open access to QoS-related user data.

particular application, a group of applications, or all detected traffic within a specific multimedia session. The 3rd party application should be notified when the provided usage monitoring thresholds have been reached.

4.3. User Data Access. The 3rd party application may need to retrieve QoS-related user data that are stored in the UDR. For example, the 3rd party application may query the UDR to obtain the QoS-related data from the user profile or its specific components, or it may browse the existing QoS-related data in user profiles in the various UDRs. The 3rd party application may add new QoS-related data in the user profile, remove, or/and modify specific QoS-related data from the repository. It is the responsibility of the Service Provider to define which QoS-related data may be modified or deleted by application providers.

The application access to QoS-related data, stored in the user profile, is depicted in Figure 4.

Functional Requirement 5. The required functions for access to QoS-related user data include the following: querying QoS data in order to retrieve the QoS parameters applied to user sessions by default; creating QoS data in order to add new QoS parameters in user profile; modifying QoS data in order to set new default QoS parameters; deleting QoS in order to erase the QoS parameters from the user profile.

Subscription/notification procedures allow the Parlay X Gateway to get notified when particular QoS data for specific user are updated in the UDR. Using functions for access to QoS-related user data, the 3rd party application can receive up-to-date information. For example, the 3rd application may request notifications about changes in QoS-related data in the user profile as shown in Figure 5. In a similar way, the 3rd party application may cancel one or several existing subscriptions.

When the data identified in subscription are changed or when the invoked subscription requests retrieval of all initial values of the referenced data, the 3rd party application is notified as shown in Figure 6.

Functional Requirement 6. To be aware of user's data changes, the 3rd party application needs functions for subscription management and means for notifications when such QoS-related events occur.

4.4. Charging Control. The charging function in PCC supports the following charging models: volume-based charging, time-based charging, time- and -volume-based charging, event-based charging, and no charging. It is possible to apply different rates and charging models (e.g., depending on the user location). The charging system selects the applicable rate based on QoS provided for the service, time of day, and

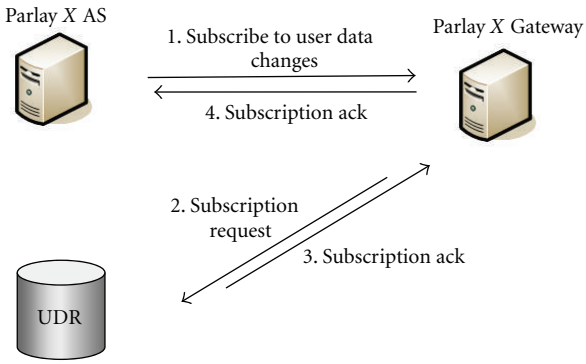

Figure 5: Subscription to QoS-related user data change.

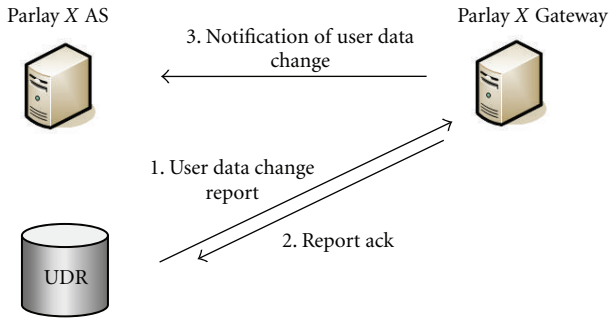

Figure 6: Notifications upon changes of user data.

so forth. In case of online charging, the charging actions are taken upon PCEF events (e.g., reauthorization upon QoS change).

Functional Requirement 7. In addition to functions for online and offline charging control, notification function is also required. To provide QoS-based charging and flow-based charging, the 3rd party application needs to be notified when some service data flows (e.g., video stream) or all service data flows (i.e., media streams of particular SIP session) have been deactivated, when the session has been terminated, or when access network has been changed.

The event types that should be reported to the 3rd party application involved in QoS management are summarized in Table 1. These event types can affect the QoS resource authorization and charging.

5. Evaluation of Parlay X Web Services Compared to Policy and Charging Control

5.1. Parlay X Application-Driven Quality of Service. The "Application-Driven Quality of Service" (ADQ) is a Parlay X Web Service that allows applications to control the QoS available on user connection. Configurable service attributes are upstream rate, downstream rate, and other QoS parameters specified by the service provider. Changes in QoS may be applied either for defined time interval or each time the user connects to the network.

Table 1: QoS-related event types.

Event type	Description
Loss/release of bearer	Loss of bearer that can result in QoS degradation (e.g., the service data flows are deactivated as a consequence). If all the bearers are lost, the application can request QoS resource release
Recovery/establishment of bearer	Recovery or establishment of a new bearer
IP-CAN change	The access network providing IP connectivity is changed, which can result in applying specific charging
Out of credit	The user credit limit is reached
Session termination	The session terminates normally
Usage report	Reports that the usage threshold provided by the 3rd party application have been reached

The ADQ ApplicationQoS interface defines operations for applying a new QoS feature to an end user connection. The ApplyQoSFeature operation is used by 3rd party application to request a default QoS feature to be set up on the end user connection, which results in a permanent change in the class of service provided over the end user connection. A default QoS feature governs the traffic flow on the end user connection whenever there are no temporary QoS features active on the connection. The ApplyQoSFeature operation is used by 3rd party application to request also a temporary QoS feature to be set up on the end user connection for a specified period of time. The ModifyQoSFeature operation is used by 3rd party application to alter the configurable service attributes (e.g., duration) of an active temporary QoS feature instance. The RemoveQoSFeature operation is used by 3rd party application to release a temporary QoS feature, which is currently active on the end user connection. Therefore, these operations provide functions required to apply, modify, and remove temporary QoS parameters (e.g., for session duration).

The ADQ Web Service enables applications to register with the service for notifications about network events that affect QoS, temporary configured on the user's connection.

The ADQ ApplicationQoSNotificationManager is used by 3rd party application to manage their registration for notifications. The startQoSNotification operation is used by 3rd party application to register their interest in receiving notifications of a specific event type(s) in context of specific end users. The stopQoSNotification operation is used by 3rd party application to stop receiving notifications by canceling an existing registration. Therefore, these operations provide functions required to manage the QoS event subscription.

The ADQ ApplicationQoSNotification interface provides the operations for notifying the Application about the impact

of certain events on QoS features that were active on the end user connection when these events occurred. The notifyQoS operation reports a network event that has occurred against end user(s) active QoS features. Therefore, this operation provides functions required to report notifications upon QoS event occurrence.

As to 3GPP TS 29.214 [25] there are indications reported over the Rx reference point by the PCRF to the P-CSCF such as recovery of bearer, establishment of bearer, IP-CAN change, out of credit, and usage report. These indications can not be forwarded to the 3rd party application by the existing definition of the enumerated type QoSEvent.

Currently, not supported by ADQ Web Service functions required for policy control include usage monitoring and resources release.

The Parlay X "Application-Driven QoS" Web service defines operations that allow retrieval of the current status of user sessions, including history list of all QoS transactions previously requested against a user session. As far as the getQoSStatus operation of the ApplicationQoS interface is used by the 3rd party application to access the currently available QoS features on a user session, it is impossible for the 3rd party application to retrieve the configured QoS features stored in the user profile. Further, if the QoS-related data in the user profile have been changed by administrative means, the 3rd party application cannot be notified.

5.2. Parlay X Payment. The Parlay X "Payment" Web Service supports payment reservations, prepaid payments, and postpaid payments. It supports payments for any content in an open, Web-like environment. When combined with ADQ Web Service, the "Payment" may be used for charging based on the negotiated QoS. The features for QoS-based charging are restricted to temporary configured QoS parameters but cannot reflect the dynamic QoS change during the session. Flow-based charging is also impossible, as far as the Parlay X "Call notification" Web Service, defined in 3GPP TS 29.199-3 [26], does not provide notifications about media addition or deletion for a particular session. Location-based charging can be applied by combination of Parlay X "Terminal Location," defined in 3GPP TS 29.199-9 [27], and "Payment" Web Services.

Table 2 shows the Parlay X Web Services support for advanced charging.

Table 2: Advanced charging functions.

Functions	Parlay X Interface	Operations
QoS-based charging	Application-Driven QoS and Payment	Notify QoS event and charge amount, refund amount
Time-of-day-based charging	Call Notification and Payment	Notify called number and charge amount, refund amount
Location-based charging	Terminal Location and Payment	Get location and charge amount, refund amount
Service flow-based charging	Audio Call and Payment	Get media for participant and charge amount, refund amount

6. Enhancement to Parlay X Web Services for PCC Support

We suggest the following interfaces to be added to the definition of "Application-Driven QoS" Web Service in order to support the PCC functionality.

6.1. New Interfaces for Usage Monitoring. The UsageMonitoringManager interface may be used by the 3rd party application to manage the usage monitoring for the accumulated usage of network resources on a per-session and user basis. The startUsageMonitoring operation may be used by the 3rd party application to set the applicable thresholds and to activate the usage monitoring. The operation parameters specify the threshold volume and whether the usage monitoring will be performed for a particular application, a group of applications, or all detected traffic belonging to a specific end user session. The stopUsageMonitoring operation may be used by the 3rd party application to cancel the usage monitoring.

The UsageMonitoringNotification interface may be used to report to the 3rd party application when threshold levels are reached. The usageMonitoringReport operation may be used to report the accumulated usage.

6.2. Enhancement to ADQ ApplicationQoS Interface. A new operation that may be defined for the ADQ ApplicationQoS interface is releaseQoSResources. The operation releases the QoS resources reserved for the user session. It may be used by the 3rd party application to release the authorized QoS resources (e.g., on receiving notification that all bearers assigned to user session are lost).

6.3. New Interfaces for Access to QoS-Related User Data. The UserDataChangeManager interface may be used by the 3rd party application to manage subscriptions for changes of user's data. The startUserDataChangeNotifications operation may be used by the 3rd party application to subscribe to receive notifications about changes in QoS-related data in user profile, made by network operator or another application. The stopUserDataChangeNotifications operation may be used by the 3rd party application to cancel the subscription for user data changes.

The UserDataChangeNotification interface may be used to report to the 3rd party application any changes in QoS-related data in the user profile. For this purpose the notifyUserDataChange operation is used.

The QoSUserData interface may be used by the 3rd party application to access to QoS-related data stored in the user profile. The interface provides operations to submit, modify, and delete QoS related data. It also provides operations to query for QoS-related data including data identifier, metadata, control data, and QoS data upload date (matching-specific criteria). The application invokes

the submitQoSData operation to submit QoS-related data into the user profile. The ADQ Web Service uploads the metadata of the QoS data to the network and the UDR stores the data. The modifyQoSData operation allows a 3rd party application to update previously submitted QoS-related and metadata. The UDR restricts modification to the submitted owner and puts the data into an invisible state until it completes the modification approval. The deleteQoSData operation allows a 3rd party application to delete QoS-related data. The readQoSData operation allows a 3rd party application to fetch the metadata of previously submitted QoS-related data. Request may include multiple data identifiers. The queryQoSData operation allows a 3rd party application to query for QoS-related data that match with specified identifiers.

6.4. Enhancement to Call Notification Functionality. We also suggest a new operation notifyMediaChange of the Call-Notification interface of the Parlay X "Call Notification" Web Service. The notifyMediaChange operation informs the 3rd party application that a media component is added to ongoing session or removed from ongoing session.

7. Mapping of Parlay X Interfaces onto Network Protocols

In order to make an adequate implementation of Parlay X "Application-Driven QoS" and "Payment" Web Services in the network, the interfaces operations have to be mapped onto messages of network control protocol.

7.1. SIP-Based Interface. The interfaces between the application server (Parlay X Gateway) and S-CSCF and between S-CSCF and P-CSCF are SIP based. SIP session information (including QoS parameters) is described by means of Session Description Protocol (SDP) and is transferred within the SIP message body. The initial request is sent as SIP INVITE message. The SIP re-INVITE message is used for modification of established session. QoS-related information about SIP session is transferred by INFO message. The management of the subscription to QoS-related events and notifications about QoS-related events are provided by means of SIP SUBSCRIBE/NOTIFY mechanism. The initial filter criteria for application triggering are stored as a part of user data stored and are downloaded to the S-CSCF on user registration.

Table 3 shows the mapping of ADQ interfaces onto SIP signaling.

The getQoSHistory operation does not require any signaling in the network and only some actions in the Parlay X Gateway.

7.2. LAPD- and SOAP-Based Interfaces. All procedures related to querying or to deleting data from the UDR and to creating or updating data within the UDR are controlled by LDAP as specified in 3GPP TS 29.335 [24]. The subscription/notification operations related to changes in user data stored within the UDR are transferred by HTTP in SOAP envelopes. Any changes in user profile create an LAPD session. To initiate an LDAP session, the Parlay X Gateway first establishes a transport connection with the UDR and then initiates an LDAP session by sending a BindRequest message. Termination of the LDAP session is initiated by the Parlay X Gateway by sending an UnbindRequest message or by the UDR by sending a Notice of Disconnection message.

In order to allow the application to relate a number of operations such as Create, Delete, and Update and to have them performed in one unit of interaction a transaction is used.

The Parlay X Gateway makes subscription for notifications about user data changes on behalf of 3rd party application by Subscribe messages. Subscribe request messages use the HTTP Post method and contain a SOAP message envelope. Subscribe response messages are coded as HTTP response message and contain a SOAP envelope. The Parlay X Gateway is notified about changes in QoS related data in user profile by Notify messages. Notify request messages use the HTTP Post method and contain a SOAP message envelope. Notify response messages are coded as HTTP response message, and contain a SOAP message envelope.

Table 4 shows the mapping of ADQ interfaces onto LAPD signaling.

7.3. Diameter-Based Interfaces. When User Data Convergence is not supported, the Parlay X Gateway is connected to the HSS. The protocol between the Parlay X Gateway and HSS is Diameter, and the 3rd party application access to user data is through Diameter commands.

To perform any changes in user data the Parlay X Gateway opens a Diameter dialogue. All 3rd party application initiated updates in user data are reflected in the HSS through the Diameter commands Profile-Update-Request/Answer (PUR/PUA). The access to user data is provided by the Diameter commands User-Data-Request/Answer (UDR/UDA).

The Parlay X Gateway subscribes to receive notifications on behalf of the 3rd party application using Diameter commands Subscribe-Notifications-Request/Answer (SNR/

Table 3: Mapping overview of ADQ interfaces onto SIP.

ADQ interface operation	SIP message
startQoSNotification	SUBSCRIBE/200[SUBSCRIBE]
startQoSNotification	SUBSCRIBE/200[SUBSCRIBE]
notifyQoSEvent	NOTIFY/200[NOTIFY]
startUsageMonitoring	SUBSCRIBE/200[SUBSCRIBE]
stopUsageMonitoring	SUBSCRIBE/200[SUBSCRIBE]
usageMonitoringReport	NOTIFY/200[NOTIFY]
applyQoSFeature (temporary)	re-INVITE
modifyQoSFeature	re-INVITE
removeQoSFeature	re-INVITE
getQoSStatus	INFO
releaseQoSResources	BYE, 200[BYE]

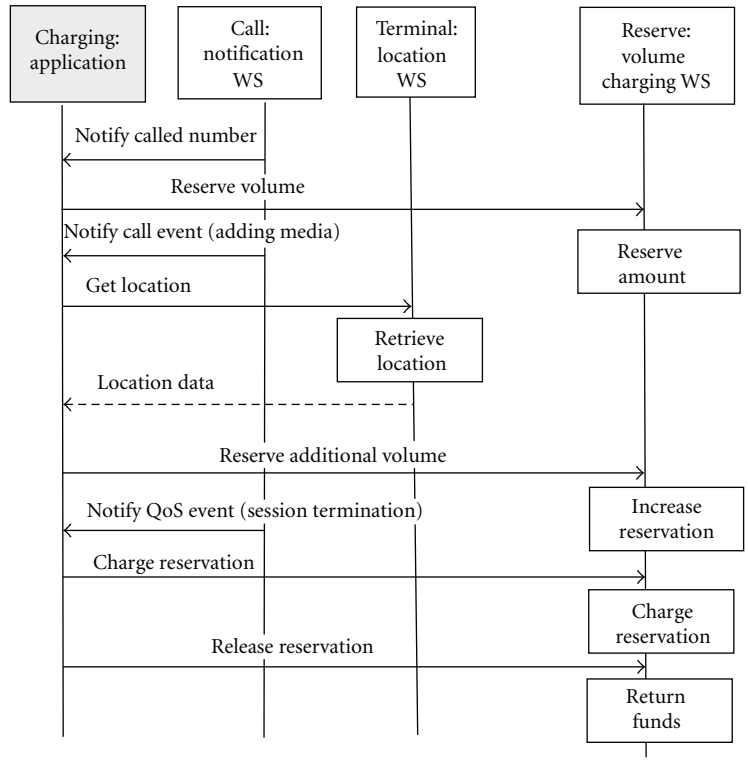

Figure 7: Use case of location-based charging.

Table 4: Mapping overview of the suggested ADQ interfaces onto UDC protocols.

ADQ interface operation	UDC protocol message
submitQoSData	LDAP AddRequest/LDAP AddResponse
modifyQoSData	LDAP ModifyRequest/LDAP ModifyResponse
deleteQoSData	LDAP DelRequest/LDAP DelResponse
readQoSData	LDAP SearchRequest
queryQoSData	LDAP SearchRequest/LDAP SearchResultEntry, SearchResultReference, and SearchResultDone
notifyUserDataChange	HTTP Post/HTTP Response
startUserDataChangeNotifications	HTTP Post/HTTP Response
stopUserDataChangeNotifications	HTTP Post/HTTP Response

SNA). Push-Notification-Request/Answer (PNR/PNA) commands are used to notify the Parlay X Gateway about events of interest.

The Rx reference point is defined between the P-CSCF and the PCRF. It is used for policy and charging control. In the context of PCC, the Diameter Authentication-Authorization-Request/Answer (AAR/AAA) commands are used to deliver SIP session information. The Re-Authorization-Request/Answer (RAR/RAA) commands report events related to QoS. The Session-Termination-Request/Answer (STR/STA) commands are used to release the resources, authorized earlier for a SIP session. The Abort-Session-Request/Answer (ASR/ASA) commands are used to provide information that all bearer resources, allocated to SIP session, are released.

8. Use Cases for Advanced Charging

To illustrate the usage of Parlay X interfaces for advanced charging we provide two use cases for service flow location-based charging and QoS-based charging.

Ann has a prepaid subscription. Shopping at a mall she decides to call Peter to invite him to a party. While discussing the details, Ann is hesitating which dress to choose and adds a video component to let Peter help her. Because of the high level of traffic load, the video stream is more expensive than usual at premises of the mall. The charging application knows that Ann is a prepaid user and, therefore, it needs to obtain permission from the online charging system (OCS). The OCS processes the credit control request and uses internal rating function to determine the rate of the

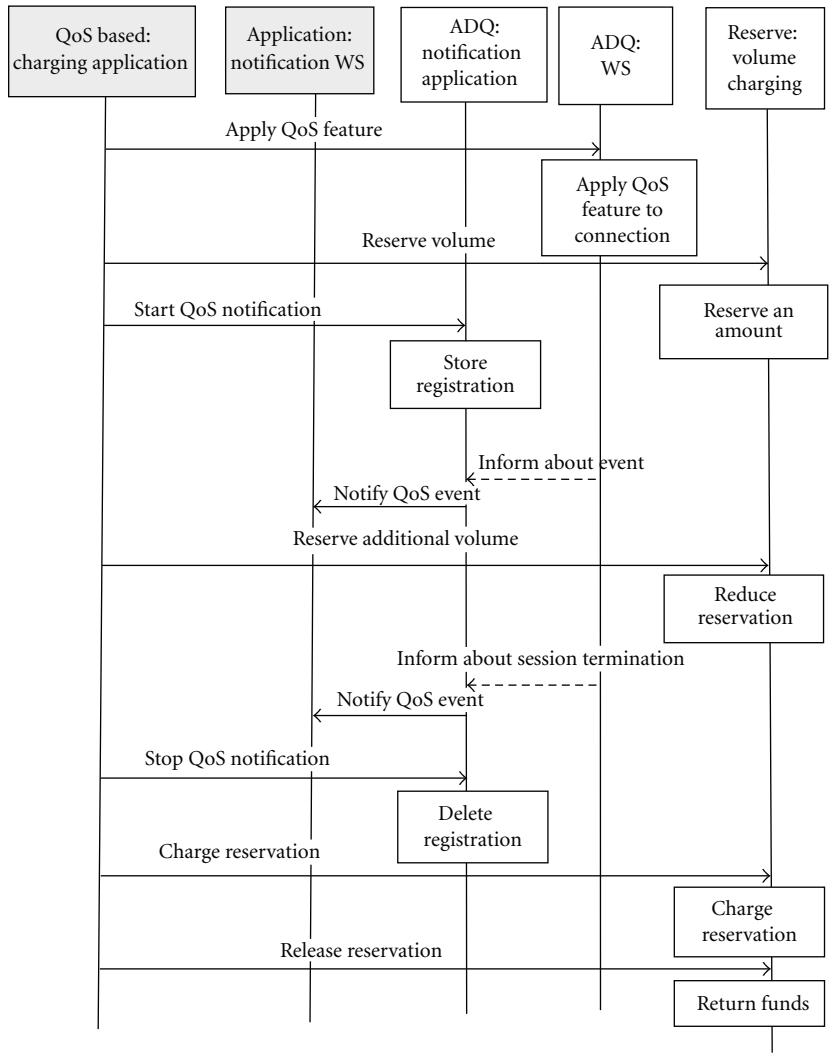

Figure 8: Use case of QoS-based charging.

desired service according to the service-specific information provided by the IMS entity if the cost was not given in the request. Then, OCS reserves an initial amount of money from Ann's account and returns the corresponding number of minutes Ann is allowed to talk. The charging application requests Ann's location when she decides to modify the media session including the video component. As Ann is in an area with scarce resources, the application determines that different charging rate has to be applied. The credit control request with charging rate information is sent to the OCS, which reserves the additional amount of money and returns the corresponding number of resources. When the minutes granted to Ann have been consumed or the service has been terminated, the OCS is informed and deducts Ann's amount from the account. The sequence diagram is shown in Figure 7.

Figure 8 shows a use case of ADQ interfaces for charging, based on the provided QoS on user session. The 3rd party application uses also the "Payment" interface and "Call Notification" interface.

In the scenario, Peter is at the stadium enjoying a football match. Peter decides to share the emotion with his friend who is away. Peter wants to send to him a video of the football match. However, the current service offering does not support the requested rate and hence it is required a temporary bit rate upgrade for the duration of the video. The QoS management application invokes the applyQoSFeature to apply new QoS parameters to the user session, specifying the higher bit rate and the duration the temporary QoS parameters should be applied. Assuming that the network allows the requested bit rate, the user's rate will be increased to the rate requested by the application for the specified duration. The application subscribes to notifications of events related to QoS available on user session. During the multimedia session the QoS goes down, so the application is notified and generates charging information based on the delivered QoS, thus correcting the requested one.

9. Conclusion

The open access to QoS management functions allows for the 3rd party applications dynamic control on QoS available on user sessions. The required functionality for open access to QoS management might be derived from the functional architecture of policy and charging control in the IP Multimedia Subsystem. The access to QoS control, gating control, flow-based charging, and user data management provides 3rd party applications with flexibility in QoS management.

So far standardized application programming interfaces do not support the entire policy and charging control functionality that network operator can expose. The evaluation of the interfaces for QoS management accessed by the 3rd party applications substantiates the need of further extension of management functions in order to provide greater flexibility in expressing communication details.

If the Parlay X approach is adopted in interface definition, besides the access to dynamic QoS control, 3rd party applications can benefit from other APIs that expose a variety of network functions. Implementation issues of the Parlay X APIs provisioning impact on the interfaces toward the network, which are left unconstrained. So, any extension of the functionality of QoS management interfaces has to be mapped onto IMS control protocols like SIP, Diameter, LDAP, and SOAP. The Parlay X Gateway has to incorporate state machines representing the 3rd party application view of interface objects that are extended with respect to the added functionality and the control protocol state machines.

The extension of the open access to QoS control adds more flexibility in resource management as far as the QoS provisioning is one of the main requirements to the IMS. Possible stakeholders that may benefit from Application-managed Quality of Service include Value Added Service providers for QoS management and 3rd party provided services that run on application servers on behalf of particular user groups.

Acknowledgment

The research is in the frame of Project DDBY02/13/17.02.2010 funded by the Bulgarian Ministry of Youth, Education and Science.

References

[1] F. Gouveia, S. Wahle, N. Blum, and T. Megedanz, "Cloud computing and EPC/IMS integration: new value-added services on demand," in *Proceedings of the 5th International ICST Mobile Multimedia Communications Conference*, 2009.

[2] S. Ouellette, L. Marchand, and S. Pierre, "A potential evolution of the policy and charging control/QoS architecture for the 3GPP IETF-based evolved packet core," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 231–239, 2011.

[3] U. Iqbal, Y. Javed, S. Rehman, and A. Khanum, "SIP-based QoS management framework for IMS multimedia services," *International Journal of Computer Science and Network Security*, vol. 10, no. 5, pp. 181–188, 2010.

[4] Y. Wang, W. Liu, and W. Guo, "Architecture of IMS over WiMAX PCC and the QoS mechanism," in *Proceedings of the IET 3rd International Conference on Wireless, Mobile and Multimedia Networks (ICWMNN '10)*, pp. 159–162, 2010.

[5] M. Jain and M. Prokopi, "The IMS 2.0 service architecture," in *Proceedings of the 2nd International Conference on Next Generation Mobile Applications, Services, and Technologies (NGMAST '08)*, pp. 3–9, September 2008.

[6] J. Yang and H. Park, "A design of open service access gateway for converged Web service," in *Proceedings of the10th International Conference on Advanced Communication Technology*, pp. 1807–1810, February 2008.

[7] R. Good and N. Ventura, "Application driven policy based resource management for IP multimedia subsystems," in *Proceedings of the 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities and Workshops (TridentCom '09)*, April 2009.

[8] R. Good, F. C. De Gouveia, N. Ventura, and T. Magedanz, "Session-based end-to-end policy control in 3GPP evolved packet system," *International Journal of Communication Systems*, vol. 23, no. 6-7, pp. 861–883, 2010.

[9] S. Musthaq, O. Salem, C. Lohr, and A. Gravey, "Policy-based QoS management for multimedia communication," 2008, http://cs.anu.edu.au/iojs/index.php/ifip/article/viewFile/13518/446.

[10] F. Zhao, L. Jiang, and C. He, "Policy-based radio resource allocation for wireless mobile networks," in *Proceedings of the IEEE International Conference Neural Networks and Signal Processing (ICNNSP '08)*, pp. 476–481, June 2008.

[11] S. G. Selvakumar, S. Paul Antony Xavier, and V. Balamurugan, "Policy based service provisioning system for WiMAX network: an approach," in *Proceedings of the International Conference on Signal Processing Communications and Networking (ICSCN '08)*, pp. 177–181, January 2008.

[12] M. Elkotob, *Autonomic resource management in IEEE 802.11 open access networks*, Dissertation, Lules University of Technology, Luleå, Sweden, 2008, http://epubl.ltu.se/1402-1757/2008/38/LTU-LIC-0838-SE.pdf.

[13] M. D. Stojanovic, S. V. B. Rakas, and V. S. Acimovic-Raspopovic, "End-to-end quality of service specification and mapping: he third party approach," *Computer Communications*, vol. 33, no. 11, pp. 1354–1368, 2010.

[14] F. Bormann, A. Braun, S. Flake, and J. Tacken, "Towards a policy and charging control architecture for online charging," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA '09)*, pp. 524–530, May 2009.

[15] H. Akhatar, "System and method for providing quality of service enablers for third party applications," Patent application number: 20090154397, 2009, http://www.faqs.org/patents/app/20090154397.

[16] M. Koutsopoulou, A. Kaloxylos, A. Alonistioti, and L. Merakos, "A platform for charging, billing, and accounting in future mobile networks," *Computer Communications*, vol. 30, no. 3, pp. 516–526, 2007.

[17] X. Duan, "Method for establishing Diameter session for packet flow based charging," 2007, http://www.freshpatents.com/%20Method-for-establishing-diameter-session-for-packet-flow-based-charging-dt20070816ptan20070189297.php.

[18] 3GPP TS 29.199-17 v9.0.0, "Open Service Access (OSA); Parlay X Web Services; Part 17: Application-driven Quality of Service (QoS), (Release 9)," 2009.

[19] 3GPP TS 29.199-6 v8.1.0, "Open Service Access (OSA); Parlay X Web Services; Part 6: Payment, (Release 9)," 2009.

[20] 3GPP TS 23.203 v11.2.0, "Policy and charging control architecture, (Release 9)," 2011.

[21] 3GPP TS 23.335 User Data Convergence (UDC), "Technical realization and information flows, (Release 9), v9.3.0," 2010.

[22] P. Calhoun, E. Guttman, G. Zorn, and J. Arkko, "RFC 3588 Diameter Base Protocol," 2003.

[23] 3GPP TS 24.229 v9.2.0, "IP Multimedia Call Control Protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP), (Release 9)," 2009.

[24] 3GPP TS 29.335 User Data Convergence (UDC), "User Data repository Access Protocol over the Ud interfaces, Release 9, v9.2.0," 2010.

[25] 3GPP TS 29.214 v11.1.0, "Policy and Charging Control over Rx reference point, (Release 9)," 2011.

[26] 3GPP TS 29.199-2 v9.0.0, "Open Service Access (OSA); Parlay X Web Services; Part 3: Call Notification, (Relase 9)," 2009.

[27] 3GPP TS 29.199-9 v9.0.0, "Open Service Access (OSA); Parlay X Web Services; Part 9: Terminal Location, (Relase 9)," 2009.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2012, Article ID 586542, 14 pages
doi:10.1155/2012/586542

Research Article

Seamless Integration of RESTful Services into the Web of Data

Markus Lanthaler[1] and Christian Gütl[1,2]

[1] *Institute for Information Systems and Computer Media, Graz University of Technology, 8010 Graz, Austria*
[2] *School of Information Systems, Curtin University of Technology, Perth WA 6102, Australia*

Correspondence should be addressed to Markus Lanthaler, markus.lanthaler@student.tugraz.at

Received 4 November 2011; Accepted 15 January 2012

Academic Editor: Nabil Tabbane

We live in an era of ever-increasing abundance of data. To cope with the information overload we suffer from every single day, more sophisticated methods are required to access, manipulate, and analyze these humongous amounts of data. By embracing the heterogeneity, which is unavoidable at such a scale, and accepting the fact that the data quality and meaning are fuzzy, more adaptable, flexible, and extensible systems can be built. RESTful services combined with Semantic Web technologies could prove to be a viable path to achieve that. Their combination allows data integration on an unprecedented scale and solves some of the problems Web developers are continuously struggling with. This paper introduces a novel approach to create machine-readable descriptions for RESTful services as a first step towards this ambitious goal. It also shows how these descriptions along with an algorithm to translate SPARQL queries to HTTP requests can be used to integrate RESTful services into a global read-write Web of Data.

1. Introduction

We live in an era where exabytes of data are produced every single year; never before in human history had we to deal with such an abundance of information. To cope with this information overload, more sophisticated methods are required to access, manipulate, and analyze these humongous amounts of data. Service-oriented architectures (SOAs) built on Web services were a first attempt to address this issue, but the utopian promise of uniform service interface standards, metadata, and universal service registries, in the form of SOAP, WSDL, and UDDI has proven elusive. This and other centralized, registry-based approaches were overwhelmed by the Web's rate of growth and the lack of a universally accepted classification scheme. In consequence, the usage of SOAP-based services is mainly limited to company-internal systems and to the integration of legacy systems. In practice, however, such a clear and crisp definition of data is rare. Today's systems integrate data from many sources. The data quality and meaning are fuzzy and the schema, if present, are likely to vary across the different sources. In very large and loosely coupled systems, such as the Internet, the gained adaptability, flexibility, and extensibility, in a transition away from strict and formal typing to simple name/value pairs or triples, outweighs the resulting loss off "correctness."

Thus, it is not surprising that RESTful services, and there especially the ones using the lightweight JavaScript Object Notation (JSON) [1] as the serialization format, are increasingly popular. According to ProgrammableWeb, 74% of the Web services are now RESTful and 45% of them use JSON as the data format [2], but, in spite of their growing adoption, RESTful services still suffer from some serious shortcomings.

The major problem is that, for RESTful services or *Web APIs,* a recently emerged term to distinguish them from their traditional SOAP-based counterparts, no agreed machine-readable description format exists. All the required information of how to invoke them and how to interpret the various resource representations is communicated out-of-band by human-readable documentations. Since machines have huge problems to understand such documentations, machine-to-machine communication is often based on static knowledge resulting in tightly coupled system. The challenge

is thus to bring some of the human Web's adaptivity to the Web of machines to allow the building of loosely coupled, reliable, and scalable systems.

Semantic annotations could prove to be a viable path to achieve that, but, while the vision of a Semantic Web has been around for more than fifteen years, it still has a long way to go before mainstream adoption will be achieved. One of the reasons for that is, in our opinion, the fear of average Web developers to use Semantic Web technologies. They are often overwhelmed by the (perceived) complexity or think they have to be AI experts to make use of the Semantic Web. Others are still waiting for a killer application making it a classical chicken-and-egg problem. A common perception is also that the Semantic Web is a disruptive technology which makes it a showstopper for enterprises needing to evolve their systems and build upon existing infrastructure investments. Obviously, some developers are also just reluctant to use new technologies. Nevertheless, we think most Web developers fear to use Semantic Web technologies for some reason or another; a phenomenon we denoted as *Semaphobia* [3]. To help developers get past this fear, and to show them that they have nothing to fear but fear itself, clear incentives along with simple specifications and guidelines are necessary. Wherever possible, upgrade paths for existing systems should be provided to build upon existing investments.

That is exactly what made the Linked Data movement so successful. It simplified the technology stack and provided clear incentives for annotating data. In consequence, it is not surprising that after being ignored by the majority of the Web developers for a long time, lightweight semantic annotations finally start to gain acceptance across the community. Facebook's Open Graph protocol, for example, was implemented in over 50,000 Web sites within the first week of its launch [4] and the current estimates are that roughly 10% of all Web pages are annotated with it.

It would just seem consequent to combine the strengths of both, REST and the Linked Data principles, but in practice they still remain largely separated. Instead of providing access to Linked Data via a RESTful service interface, current efforts deploy centralistic SPARQL endpoints or simply upload static dumps of RDF data. This also means that most current Semantic Web projects just provide read-only interfaces to the underlying data. This clearly inhibits networking effects and engagement of the crowd.

To address these issues, we developed a novel approach to semantically describe RESTful data services which allows their seamless integration into a Web of Data. We put a strong emphasis on simplicity and on not requiring any changes on the Web service itself. This should lower the entry barrier for future Web developers and provide a viable upgrade path for existing infrastructure. At the same time, the approach is extensible and flexible enough to be applicable in a wide application domain.

The reminder of the paper is organized as follows. In Section 2, we give an overview of related work. Then, in Section 3, we present the requirements and the design of SEREDASj, our approach to semantically describe RESTful services. Section 4 shows how SEREDASj can be used to integrate different RESTful services into the Web of Data, and finally, Section 5 concludes the paper and gives an overview of future work.

2. Related Work

In contrast to traditional SOAP-based services, which have agreed standards in the form of WSDL and SAWSDL [5] to be described, both, syntactically and semantically, no standards exist for RESTful services. In consequence, RESTful services are almost exclusively described by human-readable documentations describing the URLs and the data expected as input and output. There have been made many proposals to solve this issue; SA-REST [6], hRESTS [7], and WADL [8] are probably the best-known ones.

The Web Application Description Language's approach (WADL) [8] is closely related to WSDL in that a developer creates a monolithic XML file containing all the information about the service's interface. Given that WADL was specifically designed for describing RESTful services (or HTTP-based Web applications as they are called in WADL's specification), it models the resources provided by the service and the relationships between them. Each service resource is described as a request containing the used HTTP method and the required inputs as well as zero or more responses describing the expected service response representations and HTTP status codes. The data format of the request and response representations are described by embedded or referenced data format definitions. Even though WADL does not mandate any specific data format definition language, just the use of RelaxNG and XML Schema are described in the specification. The main critique of WADL is that it is complex and thus requires developers that have a certain level of training and tool support to enable the usage of WADL. This complexity contradicts the simplicity of RESTful services. In addition, WADL urges the use of specific resource hierarchies which introduce an obvious coupling between the client and the server. Servers should have the complete freedom to control their own namespace.

hRESTS (HTML for RESTful Services) [7] follows a quite different approach as it tries to exploit the fact that almost all RESTful services already have a textual documentation in the form of Web pages. hRESTS' idea is hence to enrich those, mostly already existent, human-readable documentations with so-called microformats [9] to make them machine-processable. A single HTML document enriched with hRESTS microformats can contain multiple service descriptions and conversely multiple HTML documents can together be used to document a single service (it is a common practice to split service documentations into different HTML documents to make them more digestible). Each service is described by a number of operations, that is, actions a client can perform on that service, with the corresponding URI, HTTP method, the inputs and outputs. While hRESTS offers a relatively straightforward solution to describe the resources and the supported operations, there is some lack of support for describing the used data schemas. Apart from a potential label, hRESTS does not provide any support for further machine-readable information about the inputs and outputs. Extensions like SA-REST [6] and MicroWSMO [10] address this issue.

MicroWSMO is an attempt to adapt the SAWSDL approach for the semantic description of RESTful services. It uses, just as hRESTS, on which it relies, microformats for adding semantic annotations to the HTML service documentation. Similar to SAWSDL, MicroWSMO has three types of annotations: (1) *Model*, which can be used on any hRESTS service property to point to appropriate semantic concepts; (2) *Lifting*, and (3) *Lowering*, which specify the mappings between semantic data and the underlying technical format such as XML. Therefore, MicroWSMO enables the semantic annotation of RESTful services basically in the same way in which SAWSDL supports the annotation of Web services described by WSDL.

Another approach for the semantic description of RESTful services is the before-mentioned SA-REST [6]. It relies on RDFa for marking service properties in an existing HTML service description, similar to hRESTS with MicroWSMO. As a matter of fact, it was the first approach reusing the already existing HTML service documentation to create machine-processable descriptions of RESTful services. The main differences between the two approaches are indeed not the underlying principles but rather the implementation technique. SA-REST offers the following service elements: (1) *Input* and (2) *Output* to facilitate data mediation; (3) *Lifting* and (4) *Lowering schemas* to translate the data structures that represent the inputs and outputs to the data structure of the ontology, the grounding schema; (5) *Action*, which specifies the required HTTP method to invoke the service; (6) *Operation* which defines what the service does; and (7) *Fault* to annotate errors.

In principle, a RESTful service could even be described by using WSDL 2.0 [11] with SAWSDL [5] and an ontology like OWL-S or WSMO-Lite. OWL-S (Web Ontology Language for Web Services) [12] is an upper ontology based on the W3C standard ontology OWL used to semantically annotate Web services. OWL-S consists of the following main upper ontologies: (1) the *Service Profile* for advertising and discovering services; (2) the *Service (Process) Model*, which gives a detailed description of a service's operation and describes the composition (choreography and orchestration) of one or more services; (3) the *Service Grounding*, which provides the needed details about transport protocols to invoke the service (e.g., the binding between the logic-based service description and the service's WSDL description). Generally speaking, the Service Profile provides the information needed for an agent to discover a service, while the Service Model and Service Grounding, taken together, provide enough information for an agent to make use of a service, once found [12]. The main critique of OWL-S is its limited expressiveness of service descriptions in practice. Since it practically corresponds to OWL-DL, it allows only the description of static and deterministic aspects; it does not cover any notion of time and change, nor uncertainty. Besides that, an OWL-S process cannot contain any number of completely unrelated operations [13, 14], in contrast to WSDL.

WSMO-Lite [15] is another ontology to fill SAWSDL's annotations with concrete service semantics. SAWSDL itself does not specify a language for representing the semantic models but just defines how to add semantic annotations to various parts of a WSDL document. WSMO-Lite allows bottom-up modeling of services and adopts, as the name suggests, the WSMO [16] model and makes its semantics lighter. WSMO-Lite describes the following four aspects of a Web service: (1) the *Information Model*, which defines the data model for input, output, and fault messages; (2) the *Functional Semantics*, which define the functionality, which the service offers; (3) the *Behavioral Semantics*, which define how a client has to talk to the service; (4) the *Nonfunctional Descriptions*, which define nonfunctional properties such as quality of service or price. A major advantage of WSMO-Lite is that it is not bound to a particular service description format, for example, WSDL. Consequently, it can be used to integrate approaches like, for example, hRESTS (in conjunction with MicroWSMO) with traditional WSDL-based service descriptions. Therefore, tasks such as discovery, composition, and data mediation could be performed completely independent from the underlying Web service technology.

Even though at a first glance all the above-described ideas seem to be fundamentally different from WSDL, their underlying model is still closely related to WSDL's structure. In consequence, all presented approaches heavily rely on RPC's (Remote Procedure Call) flawed [17] operation-based model ignoring the fundamental architectural properties of REST. Instead of describing the resource representations, and thus allowing a client to understand them, they adhere to the RPC-like model of describing the inputs and outputs as well as the supported operations which result in tight coupling. The obvious consequence is that these approaches do not align well with clear RESTful service design.

One of the approaches avoiding the RPC-orientation, and thus more suitable for RESTful services, is ReLL [18], the Resource Linking Language. It is a language to describe RESTful services with emphasis on the hypermedia characteristics of the REST model. This allows, for example, a crawler to automatically retrieve the data exposed by Web APIs. One of the aims of ReLL is indeed to transform crawled data to RDF in order to harvest those already existing Web resources and to integrate them into the Semantic Web. Nevertheless, ReLL does not support semantic annotations but relies on XSLT for the transformation to RDF. This clearly limits ReLL's expressivity as it is not able to describe the resource representations semantically.

There are many other approaches that allow, just as ReLL, to transform data exposed by Web APIs to RDF. In fact, large parts of the current Web of Data are generated from non-RDF databases by tools such as D2R [19] or Triplify [20] but one of the limitations of the current Semantic Web is that it usually just provides read-only interfaces to the underlying data. So, while several Semantic Web browsers, such as Tabulator [21], Oink [22], or Disco [23], have been developed to display RDF data, the challenge of how to edit, extend, or annotate this data has so far been left largely unaddressed. There exist a few single-graph editors including RDFAuthor [24] and ISAViz [25] but, to our best knowledge, Tabulator Redux [26] is the only editor that allows the editing of graphs derived from multiple sources.

To mitigate this situation, the *pushback project* [27] was initiated in 2009 (it is not clear whether this project is still active) to develop a method to write data back from RDF graphs to non-RDF data sources such as Web APIs. The approach chosen by the pushback project was to extend the RDF wrappers, which transform non-RDF data from Web APIs to RDF data, to additionally support write operations. This is achieved by a process called *fusion* that automatically annotates an existing HTML form with RDFa. The resulting *RDForm* then reports the changed data as RDF back to the pushback controller which in turn relays the changes to the RDF write wrapper that then eventually translates them into an HTTP request understandable to the Web API. One of the major challenges is to create the read-write wrappers as there are, as explained before, no agreed standards for describing RESTful services; neither syntactically nor semantically. Exposing these Web APIs as read-write Linked Data is, therefore, more an art than a science.

3. Semantic Description of RESTful Services

A machine-readable documentation of a service's interface and the data it exposes is a first step towards their (semi-) automatic integration. In this section, we first discuss the requirements for a semantic description language for RESTful services and then present SEREDASj, a novel approach to address this ambitious challenge.

3.1. Requirements. Analyzing the related work and taking into account our experience in creating RESTful services and integrating them into mashups, we derived a set of core requirements for a semantic description language.

Since the description language is targeted towards RESTful services, it clearly has to adhere to REST's architectural constraints [28] which can be summarized as follows: (1) *stateless interaction*, (2) *uniform interface*, (3) *identification of resources*, (4) *manipulation of resources through representations*, (5) *self-descriptive messages*, and (6) *hypermedia as the engine of application state*. Stateless interaction means that all the session state is kept entirely on the client and that each request from the client to the server has to contain all the necessary information for the server to understand the request; this makes interactions with the server independent of each other and decouples the client from the server. All the interactions in a RESTful system are performed via a uniform interface which decouples the implementations from the services they provide. To obtain such a uniform interface, every resource is accessible through a representation (whether the representation is in the same format as the raw source, or is derived from the source, remains hidden behind the interface) and has to have an identifier. All resource representations should be self-descriptive, that is, they are somehow labeled with their type which specifies how they are to be interpreted. Finally, the *hypermedia as the engine of application state* (HATEOAS) constraint refers to the use of hyperlinks in resource representations as a way of navigating the state machine of an application.

To be widely accepted, the approach has to be based on core Web standards. That means it should use Uniform Resource Identifiers (URIs) for identifying resources, the Hypertext Transfer Protocol (HTTP) for accessing and modifying resource representations, and the Resource Description Framework (RDF) as the unified data model for describing resources. To ease tasks such as data integration, a uniform interface to access heterogeneous data sources in a uniform and intuitive way, has to be provided as well. This, in turn, will lead to reusability and flexibility which are important aspects for the adoption of such a new approach. By having semantically annotated data, a developer could also be supported in the data integration and mediation process which is not only important in enterprise scenarios but also for the creation of mashups. All too often the required data mediation code is longer than the actual business logic. By having semantically annotated data, it is possible to integrate it (semi-) automatically with other data sources.

While all of these constraints are important when designing a RESTful service, the most important aspects for a semantic description language are how the resources can be accessed, how they are represented, and how they are interlinked. The description language should be expressive enough to describe how resource representation can be retrieved and manipulated, and what the meaning of those representations is. To integrate the service into the Semantic Web, the description language should also provide means to transform the representations in RDF triples. In order to be able to evolve systems and build upon existing infrastructure, an important requirement is that no (or just minimal) changes on the existing system are required; this implies a requirement to support partial descriptions. Last but not least, the approach should be as simple as possible to lower the entry barrier for developers and to foster its adoption.

3.2. SEREDASj. Considering the requirements described in the previous section, we designed SEREDASj a language to describe *SEmantic REstful DAta Services.* The "j" at the end should highlight that we based the approach on JSON. JSON's popularity in Web APIs is not the only reason for that.

The inherent impedance mismatch (the so-called O/X impedance mismatch) between XML, which is used in traditional SOAP-based Web services, and object-oriented programming constructs often results in severe interoperability problems. The fundamental problem is that the XML Schema language (XSD) has a number of type system constructs which simply do not exist in commonly used object-oriented programming languages such as, for example, Java. This leads in consequence to interoperability problems because each SOAP stack has its own way of mapping the various XSD-type system constructs to objects in the target platform's programming language and vice versa.

In most use cases addressed by Web services, all a developer wants to do is to interchange data—and here we are distinguishing between data interchange and document interchange. JSON was specifically designed for this: it is a lightweight, language-independent data-interchange format which is easy to parse and easy to generate. Furthermore, it

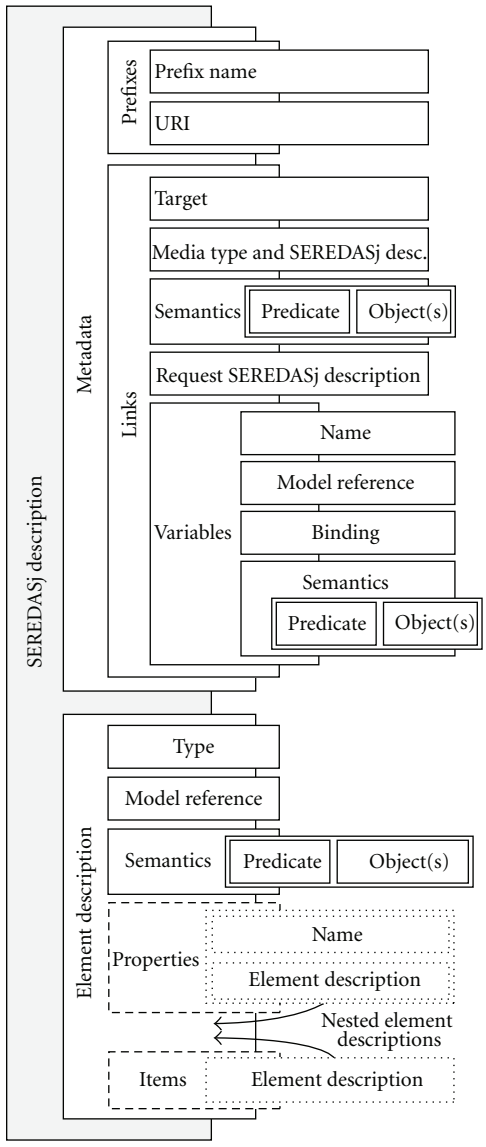

Figure 1: The SEREDASj description model.

is much easier for developers to understand and use. JSON's whole specification [1] consists of ten pages (with the actual content being a mere four pages) compared to XML where the XML Core Working group alone [29] lists *XML*, *XML Namespaces*, *XML Inclusions*, *XML Information Set*, *xml:id*, *XML Base*, and *Associating Stylesheets with XML* as standards; not even including *XML Schema Part 1* and *XML Schema Part 2*.

Summarized, JSON's simplicity, ease of integration, and raising adoption across the Web community [2] made it the first choice for our description language, but we would like to highlight that the principles of our approach are applicable to any serialization format.

To describe a RESTful service, SEREDASj specifies, similar to schemas, the syntactic structure of a specific JSON representation. Additionally, it allows to reference JSON elements to concepts in a vocabulary or ontology and to further describe the element itself by semantic annotations. Figure 1 depicts the structure of an SEREDASj description.

A description consists of metadata and a description of the structure of the JSON instance data representations it describes. The metadata contains information about the hyperlinks related to the instance data and prefix definitions to abbreviate long URIs in the semantic annotations to CURIEs [30]. The link descriptions contain all the necessary information for a client to retrieve and manipulate instance data. Additionally to the link's target, its media type and the target's SEREDASj description, link descriptions can contain the needed SEREDASj request description to create requests and semantic annotations to describe the link, for example, its relation to the current representation. The link's target is expressed by link templates where the associated variables can be bound to an element in the instance data and/or linked to a conceptual model, for example, a class or property in an ontology. The link template's variables can be further described by generic semantic annotations in the form of predicate-object pairs. The links' SEREDASj request description allows a client to construct the request bodies used in `POST` or `PUT` operations to create or update resources.

The description of the structure of instance representations (denoted as element description in Figure 1) defines the JSON data type(s) as well as links to conceptual models. Furthermore, it may contain semantic annotations to describe an element further and, if the element represents either a JSON object or array, a description of its properties, respectively, items in term of, again, an element description. The structure of the JSON instance arises out of nested element descriptions. To allow reuse, the type of an element description can be set to the URI of another model definition or another part within the current model definition. To address different parts of a model, a slash-delimited fragment resolution is used. In Listing 1, for instance, `event.json#properties/enddate` refers to the end date property defined by the SEREDASj document `event.json`.

In order to better illustrate the approach, a simple example of a JSON representation and its corresponding SEREDASj description are given in Listing 1. The example is a representation of an event and its performers from an imaginary event site's API. Without annotations, the data cannot be understood by a machine and even for a human it is not evident that a performer's ID is in fact a hyperlink to a more detailed representation of that specific performer. SEREDASj solves those problems by describing all the important aspects of such a representation. In consequence, it is not only possible to extract the hyperlinks, but also to create a human-readable documentation of the data format (as shown in [3]) and to translate the JSON representation to an RDF representation.

The SEREDASj description in Listing 1 contains two link definitions. The first one specifies the link to the performers' representations via their ID. It uses a URI template whose variable is bound to `#properties/performers/id`. This link definition also shows how further semantic annotations can be used; this is described in detail in Section 4.1. The second link specifies a search interface and is thus not

Instance Data
http://example.com/event/e48909

```
{
  "id": "e48909",
  "name": "Dick Clark's New Year's Rockin' Eve",
  "startdate": "2011-12-31",
  "enddate": "2012-01-01",
  "performers": [
    { "id": "p84098", "name": "Lady Gaga",
        "birthdate": "1986-03-28" }
  ]
}
```

SEREDASj Description
http://example.com/models/event.json

```
{
  "meta": {
    "prefixes": {
      "owl": "http://www.w3.org/2002/07/owl#",
      "so": "http://schema.org/",
      "ex": "http://example.com/onto#",
      "iana": "http://www.iana.org/link-relations/"
    },
    "links": {
      "/person/{id}": {
        "mediaType": "application/json",
        "seredasjDescription": "person.json",
        "semantics": {
          "owl:sameAs": "<#properties/performers>"
        },
        "variables": {
          "id": {
            "binding": "#properties/performers/id",
            "model": "[ex:id]"
          }
        },
        "requestDescription": "person-createupdate.json"
      },
      "/events/search{?query}": {
        "mediaType": "application/json",
        "seredasjDescription": "eventlist.json",
        "semantics": {
          "[iana:relation]": "[iana:search]" },
        "variables": {
          "query": { "model": "[so:name]" }
        }
      }
    }
  },
  "type": "object",
  "model": "[so:Event]",
  "properties": {
    "id": {
      "type": "string", "model": "[ex:id]" },
    "name": {
      "type": "string", "model": "[so:name]" },
    "startdate": {
      "type": "string", "model": "[so:startDate]" },
    "enddate": {
      "type": "string", "model": "[so:endDate]" },
```

Listing 1: Continued.

```
      "performers": {
        "type": "array",
        "model": "[so:performers]",
        "items": {
          "type": "object", "model": "[so:Person]",
          "properties": {
            "id": {
              "type": "string", "model": "[ex:id]" },
            "name": {
              "type": "string", "model": "[so:name]" },
            "birthdate": {
              "type": "string", "model": "[so:birthDate]" }
          }
        }
      }
    }
  }
```

Listing 1: An exemplary JSON representation and its corresponding SEREDASj description.

bound to any element in the instance data; instead, the variable's model reference is specified. Again, this link is semantically annotated so that an agent will know that this link specifies a search interface. These semantic annotations allow developers to implement smarter clients understanding the relationships of resources and thus following REST's hypermedia as the engine of application state constraint.

The following description of the representation's structure basically maps the structure to the ontology defined by schema.org [31]. The mapping strategy is similar to the table-to-class, column-to-predicate strategy of current relational database-to-RDF approaches [32]; JSON objects are mapped to classes, all the rest to predicates. By reusing schema.org's ontology wherever possible, the developer is able to exploit the already available human-readable descriptions for the various elements and generate completely automatically a human-readable documentation.

SEREDASj descriptions do not have to be complete, that is, they do not need to describe every element in all details. If an unknown element is encountered in an instance representation, it is simply ignored. This way, SEREDASj allows forward compatibility as well as extensibility and diminishes the coupling. In this context, it should also be emphasized that a SEREDASj description does not imply a shared data model between a service and a client. It just provides a description of the service's representations to ease the mapping to the client's data model.

4. Seamless Integration of RESTful Services into a Web of Data

Currently mashup developers have to deal with a plethora of heterogeneous data formats and service interfaces for which little to no tooling support is available. RDF, the preferred data format of the Semantic Web, is one attempt to build a universal applicable data format to ease data integration, but, unfortunately, current Semantic Web applications mostly provide just read-only interfaces to their underlying data. We believe it should be feasible to standardize and streamline the mashup development process by combining technologies from, both, the world of Web APIs and the Semantic Web. This would, in the first place, result in higher productivity which could subsequently lead to a plethora of new applications. Potentially it could also foster the creation of mashup editors at higher levels of abstraction which could, hopefully, even allow non-technical experts to create mashups fulfilling their situational needs.

Based on SEREDASj which we introduced in the previous section, we would like to propose a new reference model for integrating traditional Web service interfaces into a global read-write graph of data. Figure 2 shows the architecture of our approach.

We broadly distinguish between an application-specific (at the top) and an application-independent layer (at the bottom). The application-independent layer at the bottom is used as a generic data access layer. It separates the application and presentation logic from the common need to manage and manipulate data from a plethora of different data sources. This separation of concerns should result in better reusability and increased development productivity.

Data from JSON-based Web services described by SEREDASj are translated into RDF data and stored along with data from native RDF sources such as SPARQL endpoints, static RDF dumps, or RDF embedded in HTML documents in a local triple store. This unification of the data format is the first step for the integration of these heterogeneous data sources. We use RDF because it reflects the way data is stored and interlinked on the Web, namely, in the form of a graph. The fact that it is schema-free and based on triples makes it the lowest common denominator for heterogeneous data sources, flexible, and easily evolvable. In addition to acting as a data integration layer, this local triple store is also used for caching the

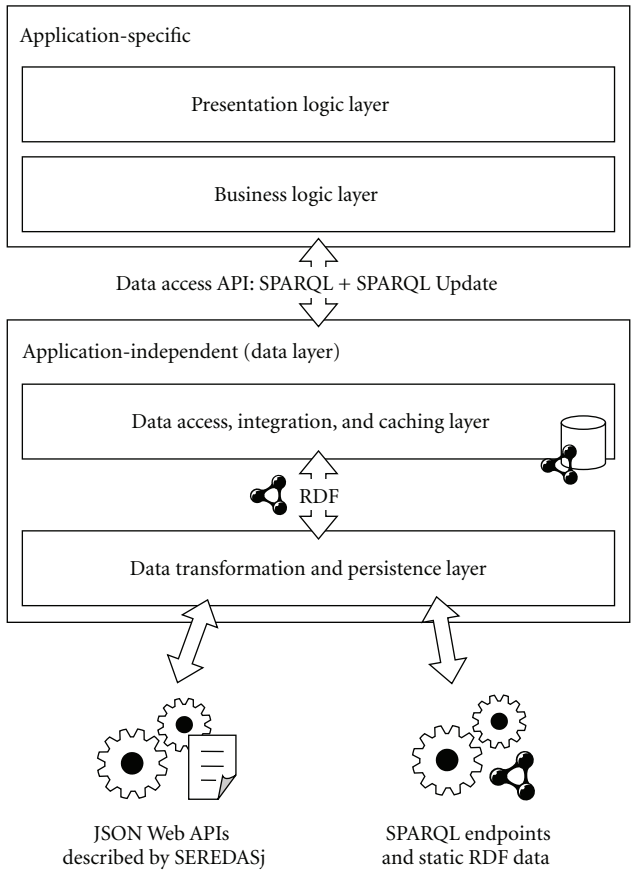

Figure 2: A reference model for integrating Web APIs into the Web of Data.

data which is a fundamental requirement in networked applications. Furthermore, centralized processing is much more efficient than federated queries and the like. Just look at, for example, Google's centralized processing compared to federated database queries and please keep in mind that we are not arguing against achievable speed increases by parallelization.

All data modifications are passed through the data access and persistence layer and will eventually be transferred back to the originating data source. The interface connecting the data access layer and the business logic layer has to be aware of which data can be changed and which cannot since some data sources or part of the data representations might be read-only. Depending on the scenario, a developer might choose to include a storage service (either a triple store or a traditional Web API) which allows storing changes even to immutable data. It is then the responsibility of the data integration layer to "replace" or "overwrite" this read-only data with its superseding data. Keeping track of the data's provenance is thus a very important feature.

In order to decouple the application-specific layer from the application-independent data layer, the interface between them has to be standardized. There exist already a standard and a working draft for that, namely, SPARQL [33] and SPARQL Update [34]. We reuse them in order to build our approach upon existing work. Of course, an application developer is free to add another layer of abstraction on top of that—similar to the common practice of using an O/R mapper (object-relational mapper) to access SQL databases.

While this three-tier architecture is well known and widely used in application development, to our best knowledge it has not been used for integrating Web services into the Semantic Web. Furthermore, this integration approach has not been used to generalize the interface of Web services. Developers are still struggling with highly diverse Web service interfaces.

4.1. Data Format Harmonization. Translating SEREDASj described JSON representations to RDF triples, the first step for integrating them into the Linked Data Cloud, is a straightforward process. The translation starts at the root of the JSON representation and considers all model references of JSON objects and tuple-typed arrays to be RDF classes, while all the other elements' model references are considered to be RDF predicates where the value of that element will be taken as object. If a representation contains nested objects, just as the example in Listing 1, a slash-delimited URI fragment is used to identify the nested object. Semantic annotations in the form of the `semantics` property, as the one shown in the performer's link in Listing 1, contain the predicate and the object. The object might point to a specific element in the SEREDASj description and is eventually translated to a link in the instance data.

The automatic translation of the example from Listing 1 to RDF is shown in Listing 3. The event and its performers are nicely mapped to schema.org ontology. For every array item, a new object URI is created by using a slash-delimited URI fragment. Eventually, those URIs are mapped to the performer's "real" URI by the link's semantic annotation. Please note that the query link is not included in the RDF representation. The reason for this is that the query variable is not bound to any instance element and thus its value is unknown. In consequence, the translator is unable to construct the URI.

4.2. Integration with Other Data Sources. As explained in the previous section, the conversion to RDF is a first step towards integration of data from different sources. To be fully integrated, the data from all sources eventually has to use the same semantic annotations, that is, the same vocabulary and the same identifiers. Traditionally, this homogenization has been done in an imperative way by writing data mediation code. The Semantic Web technology stack on the other hand embraces the inevitable heterogeneity and provides means to address this issue in a declarative way by creating new knowledge in the form of, for example, schema or identifier mappings. By studying the contents of data and the relationships between different data items, it is sometimes possible to infer (semi-) automatically that two seemingly different items are really the same.

It is straightforward to integrate the data from our example in Listing 3 with data about Lady Gaga stored in, for example, DBpedia (a project aiming to extract structured content from the information contained in Wikipedia). All we have to do is to map some of schema.org concepts and

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dbpprop: <http://dbpedia.org/property/>

SELECT ?s
WHERE {
  ?s foaf:name ?name;
    dbpprop:birthDate ?dob.
  FILTER(str(?name) = "Lady Gaga").
  FILTER(str(?dob) = "1986-03-28") }
```

Listing 2: SPARQL query to find Lady Gaga's identifier in DBpedia.

```
@base <http://example.com/event/e48909>.

@prefix rdf:
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix so: <http://schema.org/>.
@prefix ex: <http://example.com/onto#>.

<#> rdf:type so:Event.
<#> ex:id "e48909".
<#> so:name "Dick Clark's New Year's Rockin' Eve".
<#> so:startDate "2011-12-31".
<#> so:endDate "2012-01-01".
<#> so:performers <#performers/0>.

<#performers/0> rdf:type so:Person.
<#performers/0> ex:id "p84098".
<#performers/0> so:name "Lady Gaga".
<#performers/0> so:birthDate "1986-03-28".

<http://example.com/person/p84098> owl:sameAs
  <#performers/0>.
```

Listing 3: The example in Listing 1 translated to RDF.

our local identifier to concepts and Lady Gaga's identifier in DBpedia. Schema mappings are already provided by DBpedia (http://mappings.dbpedia.org/) so all we have to do is to find DBpedia's identifier and map it to our local identifier. An inference engine could do this easily by running the query shown in Listing 2 at DBpedia's SPARQL endpoint. The result is the URI we are looking for: `http://dbpedia.org/resource/Lady_Gaga`. After mapping that URI to our local identifier by using OWL's `sameAs` concept, we can easily query all the data about Lady Gaga from DBpedia as it would be part of our Web service; the data layer in Figure 2 is responsible to take care of all the necessary details.

4.3. Storing Changes Back to the Source. Just as DBpedia, a big part of the current Semantic Web consists of data transformed from Web APIs or relational databases to RDF or by data extracted from Web sites. In consequence, the vast majority of the current Semantic Web is just read-only, that is, changes cannot be stored back to the original source. Thus, in this section, we will show how SEREDASj allows data to be updated and transferred back to the originating

Web service (obviously we are not able to update static Web pages).

For the following description, we assume that all data of interest and the resulting Web of interlinked SEREDASj descriptions have already been retrieved (whether this means crawled or queried specifically is irrelevant for this work). The objective is then to update the harvested data or to add new data by using SPARQL Update.

SPARQL Update manipulates data by either adding or removing triples from a graph. The `INSERT DATA` and `DELETE DATA` operations add, respectively, remove a set of triples from a graph by using concrete data (no named variables). In contrast, the `INSERT` and `DELETE` operations also accept templates and patterns. SPARQL has no operation to change an existing triple as triples are considered to be binary: the triple either exists or it does not. This is probably the biggest difference to SQL and Web APIs and complicates the translation between an SPARQL query and the equivalent HTTP requests to interact with a Web service.

4.4. Translating Insert Data and Delete Data. In regard to a Web service, an `INSERT DATA` operation, for example, can either result in the creation of a new resource or in the manipulation of an existing one if a previously unset attribute of an existing resource is set. The same applies for a `DELETE DATA` operation which could just unset one attribute of a resource or delete a whole resource. A resource will only be deleted if all triples describing that resource are deleted. This mismatch or, better, conceptual gap between triples and resource attributes implies that constraints imposed by the Web service's interface are transferred to SPARQL's semantic layer. In consequence, some operations which are completely valid if applied to a native triple store are invalid when applied to a Web API. If these constraints are documented in the interface description, that is, the SEREDASj document, in the form of semantic annotations, a client is able to construct valid requests, respectively, to detect invalid requests and to give meaningful error messages. If these constraints are not documented, a client has no choice but to try and issue requests to the server and evaluate its response. This is similar to HTML forms with, and without client side form validation in the human Web.

In order to better explain the translation algorithm, and as a proof of concept, we implemented a simple event guide Web service based on the interface described in Listing 1. Its only function is to store events and their respective performers via a RESTful interface. The CRUD operations are mapped to the HTTP verbs `POST`, `GET`, `PUT`, and `DELETE` and no authentication mechanism is used as we currently do not have an ontology to describe this in an SEREDASj document (this is a limitation that will be addressed in future work).

The event representations can be accessed by `/event/{id}` URIs while the performers are accessible by `/person/{id}` URIs. Both can be edited by PUTing an updated JSON representation to the respective URI. New events and performers/persons can be created by POSTing a JSON representation to the collection URI. All this information as well as the mapping to the respective vocabularies is described machine-readable by SEREDASj documents.

Since SPARQL differentiates between data and template operations, we split the translation algorithm into two parts. Algorithm 1 translates SPARQL DATA operations to HTTP requests interacting with the Web service and Algorithm 2 deals with SPARQL's `DELETE/INSERT` operations using patterns and templates.

Listing 4 contains an exemplary `INSERT DATA` operation which we will use to explain Algorithm 1. It creates a new event and a new performer. The event is linked to the newly created performer as well as to an existing one.

To translate the operations in Listing 4 into HTTP requests suitable to interact with the Web service, in the first step (line 2 in Algorithm 1), all potential requests are retrieved. This is done by retrieving all SEREDASj descriptions which contain model references corresponding to classes or predicates used in the SPARQL triples; this step also takes into consideration whether an existing resource should be updated or a new one created. Since Listing 4 does not reference existing resources (`pers:p84098` in line 10 is just used as an object), all potential HTTP requests have to create new resources, that is, have to be `POST` requests. In our trivial example, we get two potential requests, one for the creation of a new event resource and a second for a new person/performer resource. These request templates are then filled with information from the SPARQL triples (line 6) as well as with information stored in the local triple store (line 7). Then, provided a request is valid (line 8), that is, it contains all the mandatory data, it will be submitted (line 9). As shown in Listing 5, the first valid request creates a new event (lines 1–3). Since the ID of the blank node `_:bieber` is not known yet (it gets created by the server), it is simply ignored. Provided the HTTP request was successful, in the next step the response is parsed and the new triples exposed by the Web service are removed from the SPARQL triples (line 11) and added to the local triple store (line 12). Furthermore, the blank nodes in the remaining SPARQL triples are replaced with concrete terms. In our example, this means that the triples in line 7–10 in Listing 4 are removed and the blank node in the triple in line 11 is replaced by the newly created `/event/e51972` URI. Finally, the request is removed from the potential requests list and a flag is set (line 13-14, Algorithm 1) signaling that progress has been made within the current `do while` iteration. If in one loop iteration, which cycles through all potential requests, no progress has been made, the process is stopped (line 18). In our example, the process is repeated for request to create a person which again results in a `POST` request (line 6–8, Listing 5). Since there are no more potential requests available, the next iteration of the `do while` loop begins.

The only remaining triple is the previously updated triple in line 11 (Listing 4), thus, the only potential request this time is a `PUT` request to update the newly created `/event/e51972`. As before, the request template is filled with "knowledge" from the local triple store and the remaining SPARQL triples and eventually processed. Since there are no more SPARQL triples to process, the `do while` loop terminates and a success message is returned

```
1 do
2   requests ← retrievePotentialRequests(triples)
3   progress ← false
4   while requests.hasNext() = true do
5       request ← requests.next()
6       request.setData(triples)
7       request.setData(tripleStore)
8       if isValid(request) = true then
9         if request.submit() = success then
10           resp ← request.parseResponse()
11           triples.update(resp.getTriples())
12           tripleStore.update(resp.getTriples())
13           requests.remove(request)
14           progress ← true
15         end if
16       end if
17   end while
18 while progress = true
19 if triples.empty() = true then
20   success()
21 else
22   error(triples)
23 end if
```

Algorithm 1: SPARQL DATA operations to Web API translation algorithm.

```
1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
3 PREFIX so: <http://schema.org/>
4 PREFIX ex: <http://example.com/onto#>
5 PREFIX pers: <http://example.com/person/>

6 INSERT DATA {
7     _:greatg a so:Event;
8             so:name "Great Gig";
9             so:startDate "2012-08-03";
10            so:performers pers:p84098;
11            so:performers _:bieber.
12    _:bieber a so:Person;
13            so:name "Justin Bieber";
14            so:gender "male";
15            so:birthDate "1994-03-01".
16 }
```

Listing 4: Examplary INSERT DATA operation.

to the client (line 20, Algorithm 1) as all triples have been successfully processed.

4.5. Translating DELETE/INSERT Operations. In contrast to the DATA-form operations that require concrete data and do not allow the use of named variables, the DELETE/INSERT operations are pattern based using templates to delete or add groups of triples. These operations are processed by first executing the query patterns in the WHERE clause which bind values to a set of named variables. Then, these bindings are used to instantiate the DELETE and the INSERT templates. Finally, the concrete deletes are performed followed by the concrete inserts. The DELETE/INSERT operations are, thus, in fact, transformed to concrete DELETE DATA/INSERT

```
1 → POST /event/
2       { "name": "Great Gig",
3       "performers": [{ "id": "p84098" }]}
4 ← 201 Created
5    Location: /event/e51972

6 → POST /person/
7       { "name": "Justin Bieber", "gender": "male",
8       "birthdate": "1994-03-01" }
9 ← 201 Created
10   Location: /person/p92167

11 → PUT /event/e51972
12      { "name": "Great Gig",
13      "performers": [ { "id": "p84098" } ,
14                    { "id": "p92167" } ] }
15 ← 200 OK
```

Listing 5: INSERT DATA operation translated to HTTP requests.

```
1 select ← createSelect(query)
2 bindings ← tripleStore.execute(select)

3 for each binding in bindings do
4   deleteData ← createDeleteData(query, binding)
5   operations.add(deleteData)
6   insertData ← createInsertData(query, binding)
7   operations.add(insertData)
8 end for

9 operations.sort()
10 translateDataOperations(operations)
```

Algorithm 2: SPARQL DELETE/INSERT operations to HTTP requests translation algorithm.

```
1 DELETE {
2  ?per so:gender ?gender.
3 }
4 INSERT {
5   ?per so:gender "female".
6 }
7 WHERE {
8   ?per a so:Person;
9     so:name "Lady Gaga";
10    so:birthDate "1986-03-28";
11    so:gender ?gender.
12 }
```

Listing 6: Examplary DELETE/INSERT operation.

DATA operations before execution. We exploit this fact in Algorithm 2 which transforms DELETE/INSERT operations to DELETE DATA/INSERT DATA operations which are then translated by Algorithm 1 into HTTP requests.

Listing 6 contains an exemplary DELETE/INSERT operation which replaces the gender of all persons whose name "Lady Gaga" and whose birth date is March 28, 1986, with "female" regardless of what it was before. This operation is first translated to a DELETE DATA/INSERT DATA operation by Algorithm 2 and then to HTTP requests by Algorithm 1.

```
1 DELETE DATA {
2   </person/p84098> so:gender "unknown".
3 }
4 INSERT DATA {
5   </person/p84098> so:gender "female".
6 }
```

Listing 7: DELETE DATA/INSERT DATA operations generated by Algorithm 2 out of Listing 6.

The first step (line 1, Algorithm 2) is to create a SELECT query out of the WHERE clause. This query is then executed on the local triple store returning the bindings for the DELETE and INSERT templates (line 2). This implies that all relevant data has to be included in the local triple store (an assumption made earlier in this work), otherwise, the operation might be executed just partially. For each of the retrieved bindings (line 3), one DELETE DATA (line 4) and one INSERT DATA (line 6) operation are created. In our example, the result consists of a single binding, namely, </person/p84098> for per and some unknown value for gender. Therefore, only one DELETE DATA and one INSERT DATA operation are created as shown in Listing 7. Finally, these operations are sorted (line 9) as deletes have to be executed before inserts and eventually translated into HTTP requests (line 10) by Algorithm 1.

In many cases, just as demonstrated in the example, a DELETE/INSERT operation will actually represent a replacement of triples. Thus, both, the DELETE DATA and the INSERT DATA operation are performed locally before issuing the HTTP request. This optimization reduces the number of HTTP requests since attributes do not have to be reset before getting set to the desired value. In our example this consolidates the two PUT requests to one.

5. Conclusions and Future Work

In this paper, we presented SEREDASj, a new approach to describe RESTful data services. In contrast to previous approaches, we put strong emphasis on simplicity to lower the entry barrier. Web developers can use tools and knowledge they are mostly already familiar with. Since SEREDASj does not require any changes on the described Web service, it provides a viable upgrade path for existing infrastructure. We also introduced two algorithms to translate SPARQL Update operations to HTTP requests interacting with an SEREDASj-described Web API. This creates a standardized interface which not only increases the developer's productivity but also improves code reusability.

A limitation of the current proposal is that it is restricted to resources represented in JSON; no other media types are supported at the moment. In future work, support should be extended to other formats such as, for example, XML. Potentially, this could be done by mapping XML representations to JSON as there are already promising approaches such as the JSON Markup Language (JsonML) [15] to do so. This would allow to transparently support XML representations without changing the current approach. Similarly, URI templates could be used to support the popular *application/x-www-form-urlencoded* media type.

In future work, we would also like to create a tool suite for developers to support the creation of SEREDASj descriptions and, if needed, the automatic creation of domain ontologies with techniques similar to the ones used to create domain ontologies from relational databases [32]. Moreover, we would like to research aspects such as service discovery and composition which includes issues like authentication that might require the creation of a lightweight ontology to be described.

References

[1] The application/json Media Type for JavaScript Object Notation (JSON), Request for Comments 4627, Internet Engineering Task Force (IETF), 2006.

[2] T. Vitvar and J. Musser, "ProgrammableWeb.com: statistics, trends, and best practices," in *Proceedings of the 4th International Workshop on Web APIs and Services Mashups*, 2010.

[3] M. Lanthaler and C. Gütl, "A semantic description language for RESTful data services to combat Semaphobia," in *Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies (DEST '11)*, pp. 47–53, IEEE, 2011.

[4] S. L. Huang, "After f8—resources for building the personalized Web," Facebook Developer Blog, 2010, http://developers.facebook.com/blog/post/379.

[5] Semantic Annotations for WSDL and XML Schema (SAWSDL), W3C Recommendation, 2007.

[6] J. Lathem, K. Gomadam, and A. P. Sheth, "SA-REST and (S)mashups: adding semantics to RESTful services," in *Proceedings of the International Conference on Semantic Computing(ICSC '07)*, pp. 469–476, IEEE, September 2007.

[7] J. Kopecký, K. Gomadam, and T. Vitvar, "hRESTS: an HTML microformat for describing RESTful Web services," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI '08)*, pp. 619–625, 2008.

[8] M.J. Hadley, Web Application Description Language (WADL), 2009.

[9] R. Khare and T. Çelik, "Microformats: a pragmatic path to the semantic web, 2006," Tech. Rep. 06-01, CommerceNet Labs, Palo Alto, CA, USA, http://wiki.commerce.net/wikiima-ges/e/ea/CN-TR-06-01.pdf.

[10] J. Kopecký and T. Vitvar, D38v0.1 MicroWSMO: Semantic Description of RESTful Services, 2008, http://wsmo.org/TR/d38/v0.1/20080219/d38v01_20080219.pdf.

[11] Web Services Description Language (WSDL) Version 2.0, W3C Recommendation, 2007.

[12] OWL S: Semantic Markup for Web Services, W3C Member Submission, 2004, http://www.w3.org/Submission/OWL-S/.

[13] M. Klusch, "Semantic web service description," in *CASCOM: Intelligent Service Coordination in the Semantic Web*, M. Schumacher, H. Schuldt, and H. Helin, Eds., pp. 31–57, Birkhäuser, Basel, Germany, 2008.

[14] R. Lara, D. Roman, A. Polleres, and D. Fensel, "A conceptual comparison of WSMO and OWL-S," in *Proceedings of the European Conference on Web Services (ECOWS '04)*, vol. 3250, pp. 254–269, Erfurt, Germany, 2004.

[15] JSON Markup Language (JsonML), 2011, http://jsonml.org/.

[16] D. Roman, U. Keller, H. Lausen, and J. D. Bruijn, "Web service modeling ontology," *Applied Ontology*, vol. 1, no. 1, pp. 77–106, 2005.

[17] J. Waldo, G. Wyant, A. Wollrath, and S. Kendall, "A note on distributed computing," Tech. Rep., Mountain View, Calif, USA, 1994.

[18] R. Alarcón and E. Wilde, "Linking data from RESTful services," in *Proceedings of the 3rd Workshop on Linked Data on the Web*, 2010.

[19] C. Bizer and R. Cyganiak, "D2R server—publishing relational databases on the Semantic Web," in *proceedings of the 5th International Semantic Web Conference (ISWC '06)*, 2006.

[20] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller, "Triplify—lightweight linked data publication from relational databases," in *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*, pp. 621–630, 2009.

[21] T. Berners-Lee, Y. Chen, L. Chilton et al., "Tabulator: exploring and analyzing linked data on the semantic web," in *3rd International Semantic Web User Interaction Workshop (SWUI '06)*, 2006.

[22] O. Lassila, "Browsing the Semantic Web," in *Proceedings of the 5th International Workshop on Semantic (WebS '06)*, pp. 365–369, 2006.

[23] C. Bizer and T. Gauß, Disco—Hyperdata Browser, http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/.

[24] D. Steer, RDFAuthor, http://rdfweb.org/people/damian/RDFAuthor/.

[25] E. Pietriga, IsaViz: a visual authoring tool for RDF, http://www.w3.org/2001/11/IsaViz/.

[26] T. Berners-Lee, J. Hollenbach, K. Lu, J. Presbrey, E. Pru d'ommeaux, and M.M. Schraefel, "Tabulator Redux: writing into the semantic web," Tech. Rep. ECSIAM-eprint14773, University of Southampton, Southampton, UK, 2007.

[27] pushback—Write Data Back From RDF to Non-RDF Sources, http://www.w3.org/wiki/PushBackDataToLegacySources.

[28] R.T. Fielding, *Architectural styles and the design of network-based software architectures*, Ph.D. dissertation, Department of Information and Computer Science, University of California, Irvine, Calif, USA, 2000.

[29] XML Core Working Group Public Page—Pubblications, XML Core Working Group, 2011, http://www.w3.org/XML/Core/#Publications.

[30] CURIE syntax 1.0: a syntax for expressing compact URIs, W3C Working Group note. W3C, 2010, http://www.w3.org/TR/curie/.

[31] Google Inc., Yahoo Inc., and Microsoft Corporation., Schema.org, http://www.schema.org/.

[32] F. Cerbah, "Learning highly structured semantic repositories from relational databases: the RDBToOnto tool," in *Proceedings of the 5th European Semantic Web Conference (ESWC '08)*, pp. 777–781, Springer, 2008.

[33] SPARQL Query Language for RDF. W3C Recommendation, 2008, http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/.

[34] SPARQL 1.1 Update. W3C Working Draft, 2011, http://www.w3.org/TR/2011/WD-sparql11-update-20110512/.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2011, Article ID 238683, 7 pages
doi:10.1155/2011/238683

Research Article

A Framework for Automatic Web Service Discovery Based on Semantics and NLP Techniques

Asma Adala, Nabil Tabbane, and Sami Tabbane

Multimedia Mobile Radio Networks Research Unit (MEDIATRON), Higher School of Communication of Tunis (Sup'Com), University of Carthage, City of Communication Technologies, 2083 Ariana, Tunisia

Correspondence should be addressed to Asma Adala, asma.adala@gmail.com

Received 4 November 2011; Accepted 30 December 2011

Academic Editor: Tai Hoon Kim

As a greater number of Web Services are made available today, automatic discovery is recognized as an important task. To promote the automation of service discovery, different semantic languages have been created that allow describing the functionality of services in a machine interpretable form using Semantic Web technologies. The problem is that users do not have intimate knowledge about semantic Web service languages and related toolkits. In this paper, we propose a discovery framework that enables semantic Web service discovery based on keywords written in natural language. We describe a novel approach for automatic discovery of semantic Web services which employs Natural Language Processing techniques to match a user request, expressed in natural language, with a semantic Web service description. Additionally, we present an efficient semantic matching technique to compute the semantic distance between ontological concepts.

1. Introduction

A lot of web services are being offered nowadays, and this trend is going to continue in the future. A demand increases consequently for an automatic discovery framework of services that are highly relevant to user requirements.

The widespread adoption of Web services is enabled by a set of flexible and extensible XML-based standards such as WSDL [1], UDDI [2], and SOAP [3]. However, these current XML-based specifications provide only syntactical descriptions of the functionality provided by Web services and therefore still require human interaction especially during discovery process. Thus, a more reliable and effective Web service discovery approach, that is suitable for automatic processing, is needed.

The Semantic Web [4] vision has encouraged researchers to enrich existing Web services descriptions with machine-interpretable semantics, called semantic Web services, in order to automate related Web services core tasks such as discovery, composition, selection and invocation. The objective of semantic Web service technology is to minimize the manual discovery and usage of Web services, by allowing software agents and applications to automatically identify, integrate, and execute these Web resources to achieve the user objectives.

Many approaches for automatic Web service discovery have been proposed, as discussed in Section 4. However, they present several major limitations. First, some proposed discovery frameworks are based on a user request that is expressed in a specific semantic description language like OWL-S [5], WSMO [6], or WSDL-S [7]. As a result, they require the end user to have intimate knowledge of semantic Web services and related description and implementation details which makes their usage difficult for end users. Second, the discovery scope of these approaches is often limited to some Web services that are published in a specific description standard. The most prominent semantic Web services frameworks are based on OWL-S or WSMO standards. This limitation is impractical since it expects all advertised services to have semantic tagged descriptions, especially that descriptions of the vast majority of already existing Web services are specified using Web Services Description Language (WSDL) and do not have associated semantics. Furthermore, it makes assumption about the used service description language which would limit the discovery process to specific advertised services.

Also, from the service requestor's perspective, the requestor may not be aware of all the knowledge that constitutes the domain ontology. Specifically, the service requestor may not be aware of all the terms related to the service request. As a result of which many services relevant to the request may not be considered in the service discovery process.

Another limitation of some proposed framework consists on their semantic matching approaches. In fact, both service provider and service requester use domain ontologies to build semantic service description file. The semantic matchmaker uses the domain ontologies from two sides to determine their degree of semantic match. Most of proposed approaches assume that both service provider and service requester use the same ontology domain to describe service capabilities which is not applicable in real-world scenario. To overcome this ontology heterogeneity, it is needed to utilize ontology mapping techniques to coordinate the differences between these ontologies to support interoperability.

In order to address the cited limitations of existing approaches, we first propose a discovery framework based on a user query expressed in natural language. Then, we perform query preprocessing using Natural Language Processing (NLP) techniques in order to extract keywords from the user query. Compared with formal queries, keyword-based queries have many advantages. They offer a simple syntax in terms of a list of keyword phrases and open vocabularies wherein the users can use their own words to express their information requirement. Also, keyword-based search is more familiar to the user due to its widespread usage (e.g., Search Engines, UDDI registries).

However, creating a semantic Web service discovery engine using a keyword-based approach can be a complex task. In fact, many issues should be considered in order to answer these questions.

(i) How to extract the most relevant information from a semantic Web service description?

(ii) How to match keywords from the user query with textual information from a semantic Web service description?

(iii) How to map English words to ontological concepts in order to perform semantic matching?

Secondly, our proposed framework does not make any assumptions about the description language of the advertised Web Service. In effect, a published Web service could be described in WSDL or in any semantic Web service description language like OWL-S or WSMO. Finally, to overcome the ontology heterogeneity problem, our proposed framework employs some Natural Language (NLP) techniques to extract senses from user keywords and Web service descriptions. It also contains a mapping module which converts English terms present in WordNet [8, 9] lexical database to Suggested Upper Merged Ontology (SUMO) [10].

The remainder of the paper is structured as follows. We present the related work in Section 2. In Section 3, we provide a background material that is essential to understand the presented approach. Section 4 presents in details our proposed discovery framework and its different modules. In Section 5, we present some conclusions.

2. Related Work

Many research efforts have been made to present a discovery framework for Web services. They are generally devised into syntactic-based approaches and semantic based approaches. The major differences between these two approaches are summarized in Table 1. The syntactic-based search engines are usually based on WSDL Web services descriptions published in UDDI. One example is the search eSynaps [11] engine. Seekda! [12] tries to go further, by extracting semantics from the WSDL files, which enables runtime exchange of similar services and composition of services. Seekda! has not yet searched through existing semantic Web service description files, but only has made use of the WSDL file of a Web service.

The semantic-based approaches utilize semantic description for Web services to automate the discovery process and employ the Semantic Web techniques. GODO [13], for example, is a Goal-Driven approach for searching WSMO Web services. It consists of a repository with WSMO Goals and lets users state their goal by writing a sentence in plain English. A language analyzer will extract keywords from the user sentence and a WSMO Goal will be searched based on those keywords. The WSMO Goal with the highest match will be sent to WSMX, an execution environment for WSMO service discovery and composition. WSMX will then search for a WSMO Web service that is linked to the given WSMO Goal via some WSMO Mediators and return the WSMO Web service back to the user. This approach makes good use of the capabilities of the WSMO framework, but it cannot be applied for other semantic languages like OWL-S, which do not have such goal representation elements.

Sycara et al. introduced LARKS [14] for describing agent capabilities and requests, and their matchmaking. The discovery/matching engine of the matchmaker agent is based on various filters of different complexity and accuracy which users can choose. However, the model lacks in defining how service requests will be specified by users. Also, LARKS assumes the existence of a common basic vocabulary for all users.

METEOR-S discovery [15] framework addresses the problem of discovering services in a scenario where service providers and requesters may use terms from different ontologies. Their approach relies on annotating service registries (for a particular domain) and exploiting such annotations during discovery.

3. Background

In this section, we describe some concepts definitions and methodologies utilized in our framework. We first present some semantic Web related technologies. Then, we briefly describe some Natural Language Processing (NLP) techniques utilized in our approach in order to process a user query written in natural language and Web services descriptions

Table 1: Syntactic versus semantic approaches for Web services discovery.

	Syntactic-based approaches for WS discovery	Semantic-based approaches for WS discovery
Matchmaking technique	(i) A simple keyword-based search (ii) Searching based on functional parameters (ii) Searching based on syntax	(i) Exploit the semantic representation of concepts describing a Web Service and their relations in an ontology (ii) Searching based on both functional and non-functional parameters
Advantages	(i) Simple and widely used technique. (ii) Standards like UDDI exist	(i) Minimize the manual discovery and usage of Web service by allowing software agents to automatically and dynamically discover WSs (ii) Pledge the automation of WS discovery process (iii) Effective and reliable technique
Disadvantages	(i) Do not allow retrieval of Web Services with similar functionality (ii) Not suited for automatic processing (iii) Still requires human interaction	(i) More complex technique (ii) Semantic tagging of Web services may be needed

before performing semantic matchmaking. We finally present an overview about WordNet and SUMO projects.

3.1. Ontology. An ontology is an explicit shared specification of various conceptualization in a particular domain. It plays a vital role in the semantic Web and tries to capture the semantics of a domain by deploying knowledge representation primitives, enabling a machine to understand the relationships between concepts in a domain.

Because some relations and axioms, ontology can be reasoned availably, therefore we can express the semantics of a concept by establishing the complex relationship among other concepts, attributes, and instances. Domain ontology is a detailed description of the hierarchical concepts of the field. It abstracts and conceptualizes objects, relationship, and class to be expressed as a vocabulary. The sets of glossary in the vocabulary are concepts. Ontology is a detailed description of the world's conceptualization. Domain ontology is the sets of all concepts from the domain. In the actual application, people always build domain ontology in their respective fields (e.g., travel ontology, communication ontology, and medical ontology).

3.2. Ontology Languages. A number of ontology's description languages have been proposed to address the semantic heterogeneity among Web resources and services. However, OWL is considered as a major technology for the future implementation of a Semantic Web since it is based on XML so OWL information can be easily exchanged between different types of computers using different operating systems and application languages.

The Web Ontology Language (OWL) [16] is a language to define and instantiate Web ontologies. It was formerly called DAML+OIL language. OWL ontology may include descriptions of classes, along with their related properties and instances. OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. It facilitates greater machine interpretability of Web content than that supported by XML, Resource Description Framework (RDF), and RDF Schema by providing additional vocabulary along with a formal semantics [17]. OWL has three sublanguages: OWL-Lite, OWL-DL, and OWL-Full. These three increasingly expressive sublanguages are designed for use by specific communities of implementers or users [16].

3.3. Web Service Description Languages. Traditional Web services are described using XML-based standards and published into a specific registry standard.

3.3.1. WSDL. WSDL [1] is an XML format for describing network services in abstract terms derived from the concrete data formats and protocols used for implementation. However, WSDL does not support semantic description of services. For example, it does not support the definition of logical constraints between its input and output parameters although it has the concept of input and output types as defined by XSD.

3.3.2. UDDI. UDDI [2] is a well-known Web service repository. The UDDI specification consists of a programmer's API along with an XML Schema definition of supporting data structures and messages. UDDI repositories contain information about *businesses*, *services*, and *service bindings* as well as additional metadata for categorization purposes. However, UDDI does not represent service capabilities. It uses *tModels* to provide a tagging mechanism. Searching for a service in an UDDI is performed by string matching on some defined fields. Thus, it is unsuitable for locating services on the basis of a semantic specification of their functionality.

3.4. Semantic Web Service Description Languages. Semantic Web services are services that have been enriched with machine-interpretable semantics. Semantic description aims to enhance the integration and Web service discovery by utilizing the machine readable constructs of the representation.

Several standards have been proposed for creating semantic Web services. Each one of them is having their own strength and can be used in a specific situation. Some of the popular languages are described as follows.

3.4.1. OWL-S. OWL-S [5] is an OWL-based Web service ontology, which supplies Web service providers with a core set of markup language, constructs for describing the properties,

and capabilities of their Web services in unambiguous and computer interpretable form. An OWL-S description is composed of three parts which are Service Profile, Service Model, and Service Grounding. The Service profile describes service capabilities and it is the part used in the discovery process. The Service Model describes how the service works (internal processes), and the Service Grounding specifies the details of how the service can be accessed.

3.4.2. WSMO. WSMO [6] provides a conceptual framework and a formal language to describe all relevant aspects of Web services to facilitate the automation of service discovery using semantics. The overall structure of WSMO is divided into four main elements [6].

(i) Ontologies: provides the terminology used by other WSMO elements.

(ii) Web service descriptions: describes the functional and behavioral aspects of a Web service.

(iii) Goals: represents user desires.

(iv) Mediators: aims to automatically handle interoperability problems between different WSMO elements.

3.4.3. WSDL-S. Current WSDL standard operates at the syntactic level and lacks the semantic expressivity needed to represent the requirements and capabilities of Web Services [18]. WSDL-S [7] is a lightweight approach for adding semantics to Web services. In WSDL-S, the semantic models are maintained outside of WSDL documents and are referenced from the WSDL document via WSDL extensibility elements.

3.5. NLP. Natural Language processing (NLP) [19, 20] is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages. NLP is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things. In theory, natural-language processing is a very attractive method of human-computer interaction. NLP has significant overlap with the field of computational linguistics and is often considered a subfield of artificial intelligence.

In our work, we employ some NLP techniques which are presented as follows.

(i) Word splitting: is the process of parsing concatenated text (i.e., text that contains no spaces or other word separators) to infer where word breaks exist.

(ii) Stemming: is the process for reducing inflected (or sometimes derived) words to their stem, base, or root form. For example, a stemming algorithm reduces the words "fishing", "fished", "fish," and "fisher" to the root word, "fish".

(iii) Part Of Speech (POS) tagging: is the process of marking up the words in a text (corpus) as corresponding to a particular part of speech, based on both its definition as well as its context. A POS tagger enables the identification of words as nouns, verbs, adjectives, adverbs, and so forth.

(iv) Word Sense Disambiguation (WSD): the process of identifying which sense of a word (i.e., meaning) is used in a sentence, when the word has multiple meanings (polysemy).

3.6. WordNet. WordNet [8, 9] is an electronic lexical database for the English language realized at Princeton University by George Miller's team and based on psycholinguistic theories. In WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (*synsets*), each expressing a distinct concept. Synsets are interlinked by means of conceptual semantic and lexical relations.

WordNet is of interest not only because it is a vast repository of lexical data, but also because it is so widely used. It has been leveraged for automated sense-disambiguation, term expansion in IR systems, and the construction of structure representations of document content. In fact, WordNet is so popular that it is almost considered a de facto standard in the NLP community.

3.7. SUMO. SUMO (Suggested Upper Merged Ontology) [10] is an ontology that was created at Teknowledge Corporation with extensive input from the SUO (Standard Upper Ontology) [21] mailing list and it has been proposed as a starter document for the IEEE-sanctioned SUO Working Group.

The SUMO was created by merging publicly available ontological content into a single, comprehensive, and cohesive structure.

4. Proposed Framework

In this section, we present our discovery framework presented in Figure 1. We give detailed description about our proposed keyword-based discovery approach for searching Web services which are described using a syntactic or a semantic language and advertized in a Web service registry. This search mechanism incorporates natural language processing techniques to establish a match between a user search query, containing English keywords, and a Web service description. The overall process is modeled in a sequence diagram expressed in UML (Unified Modeling Language) standard and presented in Figure 2.

4.1. Framework Architecture. Our discovery process aims to enable efficient search for appropriate Web services according to a user query. In our proposed discovery framework, we suppose that there is a set a Web services described in WSDL, OWL-S, or WSMO languages and published by services providers in a Web service registry. These descriptions are parsed and read by our system in order to extract all useful information elements for the matchmaking process. Some NLP techniques are then applied to extracted information to find useful words for next steps. As words could have different senses, a sense disambiguation is performed. In order to map each word to its corresponding concept in SUMO ontology, a WordNet/SUMO mapping is carried out. The final process in the framework is semantic matchmaking.

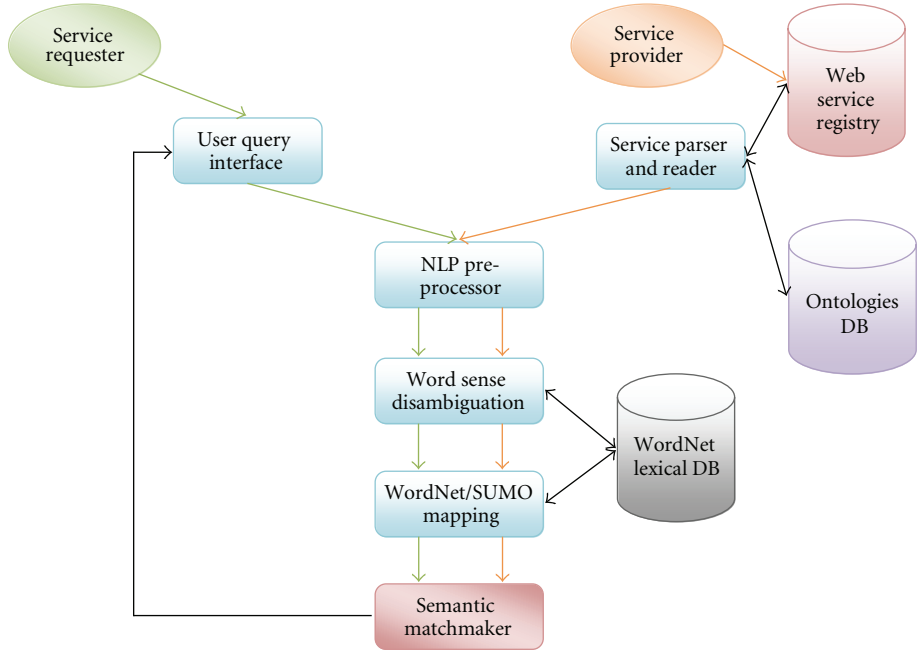

Figure 1: Automatic Web service discovery framework architecture.

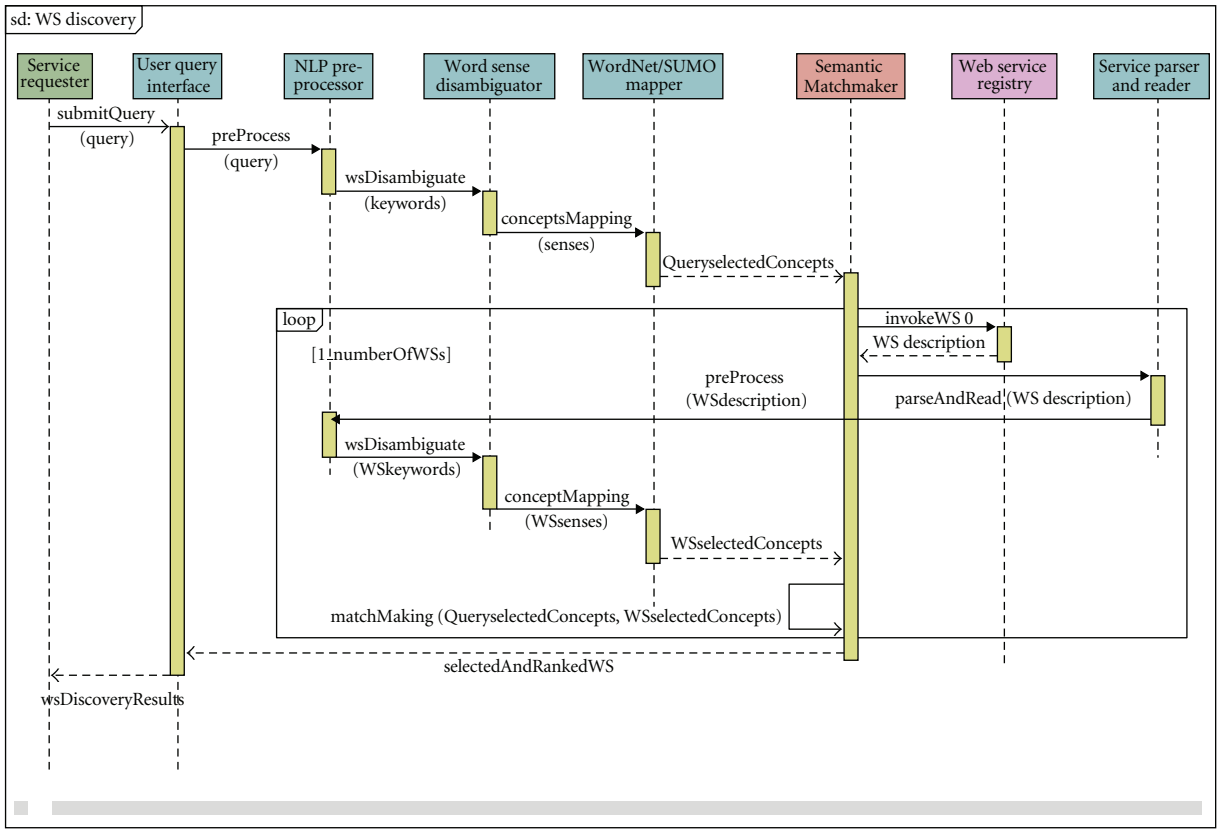

Figure 2: Sequence diagram for use case "discovery of Web services."

It is based on calculating semantic distance between concepts defined in ontology.

From the service requester point of view, our system offers a simple graphical user query interface to facilitate the discovery process. Therefore, the framework has as input a query expressed in natural language, from which useful keywords are extracted. Consequently, the overall architecture and implemented technologies are transparent to user. The user query must be also preprocessed to be matched to service description using the same processes as service description. Finally, the concepts mapped to the senses disambiguated from the search query are matched with the concepts mapped to the senses disambiguated from Web service description.

4.2. Service Parser and Reader. As presented in Sections 3.3 and 3.4, there exist many Web service description languages. For each service annotation, a different reader is needed. A reader must be able to extract elements out of a Web service description and of its used ontologies in case of semantic annotations.

In the case of OWL-S or WSMO Web service, names and nonfunctional descriptions of elements such as the capabilities (inputs/outputs), conditions, and effects of the Web service should be extracted by the service reader. After extracting concepts out of those elements, the service reader searches for their nonfunctional descriptions in the relevant ontology which is extracted from the ontologies database.

In the case of WSDL description, the service reader extracts operations parameters (all terms under <element name> and <documentation> tag).

Before extracting words from a Web service description, the description has to be parsed. Different languages can represent different syntaxes and therefore different parsers are needed. For example, for WSMO a WSML parser like WSMO4J [22] can be used. Sesame [23] and Jena [24] are examples of parsers for OWL-S.

4.3. Service and Query Preprocessor. Web service description must be preprocessed in order to transform extracted elements into useful words that could be processed later. User query must be also pre-processed to extract useful keywords from a query written in natural language. For pre-processing, some NLP techniques are utilized.

First, word segmentation is performed if needed to split a string of written language into its component words. The white space is a good approximation of a word delimiter. In the case of element names, simply splitting the words when a case transition has occurred is enough, since in most cases they are written as camel words (e.g., TravelCheckingService). To find useful words for WSD, each word in the sentences found must be tagged with the right Part-of-Speech (PoS) such as noun, verb, and adjective. Markups and punctuations are then removed. Translation of uppercase characters into lowercase is also needed. Second, all stop words are removed from extracted elements. Stemming is finally processed to transform obtained words to root words.

4.4. Word Sense Disambiguation. The Word Sense Disambiguation module establishes the context of words received from the preprocessor by extracting relevant senses. This will result in a set of senses, each representing a single meaning of a word. In general terms, WSD involves the association of a given word in service description or in user request with a definition or meaning (*sense*) which is distinguishable from other meanings potentially attributable to that word. The task therefore necessarily involves two steps: (1) the determination of all the different senses for every word and (2) a means to assign each occurrence of a word to the appropriate sense.

In our approach, we use a variant of the SSI algorithm [25] to get the senses out of a set of words as it is shown by (1). The algorithm disambiguates a word (*word*) based on a previously disambiguated set of words and their related senses. Per sense of the word (sj), a similarity with the senses from the context (sci) is calculated and the sense with the highest similarity is chosen. After that, the word and its chosen sense will be added to the context (I) and iteration will be done. This process continues until there are no ambiguous words left

$$\text{selected Sense (lex)} = \arg \max_{sj \in \text{senses (word)}} \sum_{sci \in I} sim(sj, sci). \tag{1}$$

At the start of the process, a context is not yet established. In order to disambiguate meanings of the words that can have multiple senses, one first has to find the words that have only one sense (monosemous words) to initialize the context. If all the words in the set have multiple senses (polysemous words), the least ambiguous word is chosen and for each of its senses, the algorithm is simulated as if the sense was used as the starting context. Each time a new sense is added to the context, the similarity between the new sense and the context is stored. The sense which creates the highest sum of similarity measures during its simulation is used for the context initialization.

The similarity function (*sim*) is defined in Section 3.6.

4.5. WordNet/SUMO Mapping. The mappings between WordNet and the SUMO can be regarded as a natural language index to the SUMO. It presents a tool which permits the user to enter English terms and which returns SUMO concepts that are associated with the input terms via WordNet synsets. The WordNet/SUMO mapping module offers the capability to assign the structured meanings of the SUMO to free text. In fact, all extracted senses from WSD module are matched to the equivalent concept in SUMO ontology. Thus, semantic matchmaking could be applied to user query-related concepts with service-description-related concepts.

4.6. Semantic Matchmaker. A basic step toward semantic matchmaking is to calculate the semantic distance between concepts that are defined in an ontology. In the semantic matchmaking module, we utilize a novel edge-based approach to measure the semantic distance between two ontological concepts which is presented in details in our previous

work [26]. The edge is the direct semantic relation between two concepts in the ontology. In our proposed approach, the semantic distance between two concepts is a function σ of edges weights values along the path between two concepts. An edge's weight depends on two parameters that are the depth of the parent node (super concept) in the hierarchy ($d(p)$) and the local density of the parent node ($E(p)$). This semantic distance function is defined in

$$\sigma(c_1, c_2) = \sum_{c \in \{\text{path } (c_1, c_2) \text{LS } (c_1, c_2)\}} \text{wt}(c, p(c)). \quad (2)$$

The calculation of an edge weight is expressed by

$$\text{wt}(c, p) = \left(\beta + (1 - \beta)\frac{\overline{E}}{E(p)}\right)\left(\frac{d(p) + 1}{d(p)}\right)^{\alpha}. \quad (3)$$

The semantic matchmaker has in input two sets of SUMO concepts. One set represents the user query and the other represents the service description.

Equation (4) is applied to calculate the final semantic matching degree between the two sets of concepts

$$\text{MatchD}(S_1, S_2) = \begin{cases} \underset{c_1 \in S_1}{\text{Max}}\left\{\underset{c_2 \in S_2}{\text{Min}}\{\sigma(c_1, c_2)\}\right\}, & S_1 \neq \varphi \wedge S_2 \neq \varphi, \\ 0, & S_1 = \varphi \vee S_2 = \varphi. \end{cases} \quad (4)$$

5. Conclusion

The work proposed in this paper provides an approach for automatic discovery of Web services.

We lay stress on the fact that, since users often have little knowledge about Web-service-related technologies and implementation details, a discovery framework that has a user query expressed in natural language as input is needed. Our proposed framework presents a discovery mechanism that enables Web-service-discovery-based on keywords written in natural language with no constraints about the used Web service description language. We presented a novel approach which takes advantages from keyword-based search simplicity and from Semantic web emergent technologies to automate the discovery process of Web services.

Some of our work in progress is aimed at extending our approach to service discovery, to support service invocation and workflow composition.

References

[1] W3C Web Services Description Language, http://www.w3.org/TR/wsdl.

[2] Universal Description Discovery and Integration, http://uddi.xml.org/uddi-org.

[3] SOAP Version 1.2. W3C Recommendation, 2007, http://www.w3.org/TR/soap12-part0/.

[4] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, 2001, http://www.w3.org/2001/sw.

[5] OWL-S: Semantic Markup for web services, OWL-white paper, http://www.ai.sri.com/daml/services/owl-s/1.2/overview/.

[6] J. D. Bruijn, C. Bussler, J. Domingue et al., "Web Service Modeling Ontology(WSMO)," http://www.w3.org/Submission/WSMO/.

[7] K. Sivashanmugam, K. Verma, A. Sheth, and J. Miller, "Adding semantics to web services standards," in *Proceedings of the International Conference on Web Services*, pp. 395–401, Las Vegas, Nev, USA, June 2003.

[8] G. A. Miller, "WordNet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[9] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, Mass, USA, 1998.

[10] I. Niles and A. Pease, "Linking lexicons and ontologies: mapping wordnet to the suggested upper merged ontology," in *Proceedings of the International Conference on Information and Knowledge Engineering (IKE '03)*, pp. 412–416, Las Vegas, Nev, USA, June 2003.

[11] eSynaps: eSynaps Web Service Search, 2009.

[12] Semantic Technology Institute: Seekda!, 2009.

[13] J. M. Gomez, M. Rico, F. Garcia-Sanchez, R. M. Bejar, and C. Bussler, "GODO: goal driven orchestration for semantic web services," in *Proceedings of the Workshop on Web Services Modeling Ontology Implementations*, vol. 113, CEUR Workshop Proceedings, 2004.

[14] K. Sycara, S. Widoff, M. Klusch, and J. Lu, "LARKS: dynamic matchmaking among heterogeneous software agents in cyberspace," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 2002.

[15] S. Oundhakar, K. Verma, K. Sivashanmugam, A. Sheth, and J. Miller, "Discovery of web services in a multi-ontology and federated registry environment," *International Journal of Web Services Research*, vol. 2, no. 3, pp. 1–32, 2005.

[16] Y. Alsafadi, J.-F. B. J. Barnette, S. Bechhofer et al., "OWL Web Ontology Language Guide," 2004, http://www.w3.org/TR/owl-guide/.

[17] D. L. Mcguinness and F. V. Harmelen, "OWL Web Ontology Language Overview," 2004, http://www.w3.org/TR/owl-features/.

[18] M. Herrmann, M. A. Aslam, and O. Dalferth, "Applying semantics (WSDL, WSDL-S, OWL) in service oriented architectures (SOA)," in *Proceedings of the 10th International Protégé Conference*, Budapest, Hungary, 2007.

[19] D. Jurafsky and J. H. Martin, *Speech and Langugage Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2008.

[20] N. Indurkhya and F. Damerau, *Handbook of Natural Language Processing*, CRC Press, Taylor and Francis Group, 2nd edition, 2010.

[21] Standard Upper Ontology, http://suo.ieee.org/.

[22] EU IST, FIT-IT: WSMO4J API, 2008, http://wsmo4j.sourceforge.net/.

[23] Sesame, 2009, http://www.openrdf.org/.

[24] HP Labs Semantic Web: Jena, 2008, http://jena.sourceforge.net.

[25] R. Navigli and P. Velardi, "Structural semantic interconnections: a knowledge-based approach to word sense disambiguation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1075–1086, 2005.

[26] A. Adala, N. Tabbane, and S. Tabbane, "An edge-based approach for semantic matchmaking of service capabilities," *Journal of Computer Technolgy and Application*, vol. 2, no. 8, 2011.