Steganography in Arabic Text Using Kashida Variation Algorithm (KVA)

Ammar Odeh
Computer Science & Engineering
University of Bridgeport
Bridgeport, CT 06604, USA
aodeh@bridgeport.edu

Khaled Elleithy, Miad Faezipour
Computer Science & Engineering
University of Bridgeport
Bridgeport, CT 06604, USA
{elleithy, mfaezipo}@bridgeport.edu

***Abstract*-The need for secure communications has significantly increased with the explosive growth of the internet and mobile communications. The usage of text documents has doubled several times over the past years especially with mobile devices. In this paper, we propose a new steganography algorithm for Unicode language (Arabic). The algorithm employs some Arabic language characteristics which represent extension letters. Kashida letter is an optional property for any Arabic text and usually is not popularly used. Many algorithms tried to employ this property to hide data in Arabic text. In our method, we use this property to hide data and reduce the probability of suspicions. The proposed algorithm first introduces four scenarios to add Kashida letters. Then, random concepts are employed for selecting one of the four scenarios for each round. Message segmentation principles are also applied, enabling the sender to select more than one strategy for each block of message. At the other end, the recipient can recognize which algorithm was applied and can then decrypt then message content and aggregate it. Kashida variation algorithm can be extended to other similar Unicode languages to improve robustness and capacity.**

Keywords: Steganography , Kashida, Carrier file, Information Hiding, Persian/Arabic Text, Steganalysis, Diacritics .

I. INTRODUCTION

A. Background

Steganography is a security art used to hide message inside other messages to produce Stego-object. "Stegano" means hidden and "Graptos" means writing [1]. The main idea of Steganography is to send message between two parties without any suspicions from intruders [2, 3].

When the message is traveling over an untrusted channel, we want to protect our messages. Mainly we have two scenarios:-

1. Cryptography: Hidden data meaning/content
2. Steganography: Hidden data existence

Usually the first strategy (called cryptography) is used where data is transferred from readable form (Plain text) into scribbled data (cipher text), and will then rely on transferring cipher text over the network. Receiver can decrypt cipher text to read the message by using a secret key. In the second approach, Steganography will hide data existence by using another file as carrier and a strategy to insert secret data inside it, and will then pass data through the communication channel. In this case, everyone can read the carrier file but no one can notice the hidden message [4, 5].

We can classify Steganography methods depending on the carrier file used in the algorithm. Some algorithms use image files as the carrier file. Examples of such are the least significant bit replacement algorithm (LSB) [5]. Other algorithms use audio files as a carrier secret data in the file, where signal noise can be used to hide data inside the audio file [6].

Text steganography acts as the hardest strategy to hide data inside the carrier file where text files contain the least redundant data compared to image and audio files [7, 8]. On the other hand, text steganography is the oldest hidden method used. It is said that ancient Greek (Histiaeus) shave salve hair, tattoo message in salve head, wait until his hair grows back again and then send the person with the hidden message to the other side [5].

Text steganography can be classified into three classes:-

1. Format Based: - by changing the format of carrier file, we can pass our secret message. Format strategies depend on language properties. In other words, some algorithms can be applied on specific languages and cannot be applied on other languages. Other methods can be applied on any text regardless of the carrier file language [9].

2. Random and Statistical Generation Methods: in this strategy we will generate cover text depending on statistical properties of the language. Probabilistic context-free grammar (PCFG) is the most common strategy used to produce the cover file. Other strategies rely on the statistical properties of the word used such as letter frequency and word length [9, 10].

3. Linguistic Methods: which can be divided into two groups. The first method is syntactic method which depends on some punctuation signs to hide data. The second approach can be applied by creating a synonym dictionary and then replacing the interactive word by some carrier file word to pass hidden bits [10, 11].

The algorithm presented in this paper aims to hide message inside Arabic text by using extension characters called Kashida characters to pass two bits. In addition, we use a

random algorithm to distribute the hidden bits inside the message.

B. *Main Contributions and Paper Organization*

A promising algorithm is presented in this paper. The main idea is to use Kashida in Arabic that enables us to hide bits and remove any intruder suspicions. We first introduce four scenarios to add Kashida letters and randomly select one of the four scenarios for each round. Message segmentation is applied, enabling the sender to select more than one strategy for each block of message. At the other end, the recipient can recognize which algorithm was applied and can then decrypt the message content and aggregate it.

The rest of this paper is organized as follows. In Section II we discuss previous text Steganography techniques. Kashida variation algorithm in Arabic hidden algorithm is discussed in Section III. Section IV will describe our algorithm and provide analysis regarding the proposed work. Finally, concluding remarks are offered in Section V.

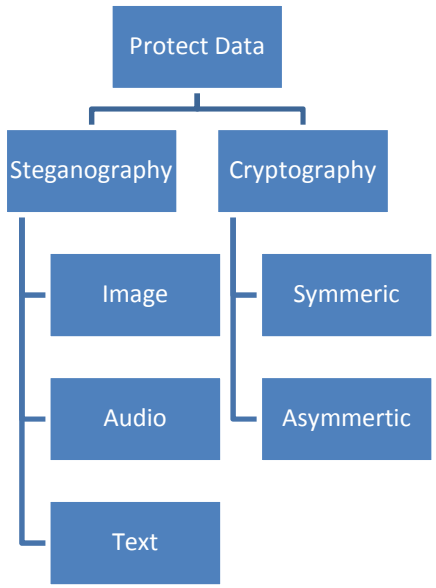

Figure 1.Hidden Data classification

II. PRIOR WORKS

Text Steganography can be classified into three strategies as shown in Figure 1. Each one of these techniques have advantages and drawbacks. In the following section, we will present some of the prior works and provide some discussions on them.

Different linguistic methods have been classified into two categories. The first one is syntax and the other one is semantic [9]. This work has been developed by creating a dictionary of synonyms, and creating a representation for each word by bit. Authors in [9] presented a novel synonym algorithm to hide data in Bahasa Melayu language, where the hidden algorithm is divided into two phases. The first step converts hidden message into binary codes by using ASCII codes. Then a synonyms file is created, where the sender and recipient must have the same word list to encrypt and decrypt the message. If the sender wants to insert a Zero, then there is no need for word replacement. Otherwise, the word from synonyms file is replaced. The same strategy will be iterated until the end of the secret message is reached. The recipient can decrypt the message by an inverting strategy and comparing if a replacement has occurred, in which case the secret code is 1.

Other similar techniques have been presented in [12]. The algorithm consists of three input sources; natural language; secret message and the key, and one output which is Stego-object. By creating lexical substitutions set and variant forms of the same word, and after the first scan, the system will recognize each word and to which set it belongs to. The lexical analyzer for Chinese language is then used to embed the correct word in the carrier file and take the context into consideration.

Line shifting techniques have been presented in [13] by vertically shifting the line. In some measurements $\{a, -a\}$ are employed so that secret data can be passed onto it. The main drawback of this strategy is that character recognition programs can detect line shifting. In addition, hidden data will be discarded by retyping the carrier document.

Authors in [14] introduced Syntactic methods using punctuations to pass the secret message by placing punctuations in proper positions to hide data. This method will not affect the meaning of the message or the data embedded inside it. On other hand, little amount of data can be embedded by using this method.

Authors in [12] employed one of the new technology techniques for Short Messaging Service (SMS). The technique relies on using message abbreviation for words, and creating a table of abbreviation and description of the message, similar to Table I.

TABLE I. SMS ABBREVIATION

Abbreviation	Description
AFAIK	As Far As I Know
ABT	About
B4	Before
BTW	By The Way
EOL	End of Lecture
CRAP	Cheap, redundant assorted products

In [15], a novel text hiding algorithm was proposed using chat properties; especially nowadays where chatting rooms are very popular. The main idea in this research was to use emoticons to hide data where tremendous number of those symbols could be used to hide data inside the file. Most of chat users use emoticons instead words nowadays. So the first step of this algorithm is to classify symbols semantically and then control the symbol order by using a secret key. The work creates 4 sets of emoticons to hide data. As an example, if the symbol is inserted at the beginning of sentence, it would pass 0, otherwise the secret bit is 1. Other strategies used in this work rely on symbol order where data can be extracted from the symbol depending on the symbol order number in its set. For example, if it is in set 4 and order 3, the secret data is

0011, and so on. A lot of advantages can be acquired from this method, as chatting programs are so popular. Moreover, some chatting programs enable users to create their own customized symbols.

New promising techniques were introduced in [16] where one of the Arabic language properties called Diacritics (Harakat) was used. This is an optional property of the language. The idea was that if we want to pass 1 we keep that Harakat, otherwise we remove it to pass zero.

In [17], the authors presented a new text steganography technique for Arabic language again by using one of Arabic characteristics; diacritics or Harakat. Usually, Haraka represent vowel sounds. Nowadays it's rarely to used, but on other hand any religious document must have it. Arabic language has 8 Diacritics where the most frequently used one is Fatha. The work uses Fatha to represent one and 0 can be represented by any of the remaining seven. In this case, 1 is the hidden bit, where the first Fatha can be found and any other Haraka before it can be removed to hide 1. One of the advantages of this method is the reusability feature where the same cover can be used for more than one hidden message. Moreover, this technique does not require complex software. On other hand, however, Harakat are rarely used nowadays.

In [18] the authors introduced a new algorithm using some of the Arabic letters which are called pointed letters (ن, ف, غ, ظ خ, ت,ث, ض) by vertical shifting of the point(s). If we want to pass a Zero, no vertical shift takes place, otherwise we pass one by vertical shifting. The work in [19] improves the vertical point shifting method by using multipoint characters only and taking into account the shifting and distance between points to pass two bits in each multipoint letter.

III. PROPOSED ALGORITHM

Some of the Arabic characters features support different Steganography algorithms. In the following section, we explore some Arabic language properties to use some of its attributes to hide large amount of data inside Unicode file (Arabic language).

1. *Writing Direction:-*

Arabic text is written from right to left. It is a unidirectional language and numbers are read and written in the same direction.

2. *Letter connectivity:-*

Most of the Arabic letters in the word are connected with the previous letter and next one. Therefore, the letter may have different shapes depending on its position in the word as shown in Figure 2.

Figure 2. Different (Mem) Letter shapes depending on its position in the word.

3. *Dotted letters:-*

Some Arabic letters have one, two and sometimes three points. These points affect the letter's pronunciation [20] as shown in Table II.

4. *Kashida letter:-*

One of Arabic language characteristic is to extend the letter, which is called Kashida letter. This is represented as (-) after joining the letter. For example (احمد), can be written as (احـمـد) where we add two Kashida. This means that we can embed 2 bits inside one word.

TABLE II. ARABIC LETTERS

Letter	Number of points
و,ه,ل,ك,ع,ط,ص,ر,د,ح,ا	0
ن,ف,غ,ظ,ض,ز,ذ,خ,ج,ب	1
ي,ق,ت	2
ش,ث	3

Therefore, if we have any word which consists of N connected letters we can embed N-1 bits inside it. If our text contains J words that contain connective letters, the number of embedded bits is:

$$Embedded = (N\text{-}1) \times J; \qquad (1)$$

In our algorithm, we exploit this property in Arabic text to hide data inside cover media. In addition, we apply four scenarios randomly to improve data privacy. This is while most of the previous algorithms apply the same procedure for the whole text, in which case they may allow Stegoanalysis to follow up the text format, which may lead to breaking the hiding algorithm.

Our proposed algorithm has four scenarios as follows:
Scenario 0:- Adding Kashida after pointed letters to encode one, otherwise, we pass zero.
Scenario 1:- Adding Kashida after non-pointed letters to encode one, otherwise, we pass zero.
Scenario 2:- Adding Kashida after letters to encode one, otherwise, we pass zero.
Scenario 3:- Adding Kashida after letters to encode zero, otherwise, we pass one.

Therefore, we will randomly use these scenarios to hide data. At the beginning, we will divide the text into blocks as shown in Figure 3. Each block will pick up a scenario from the suggested scenarios above.

Algorithm I: Segmentation and Aggregation
Input Message
Segment Message into Blocks (Block 0, Block 1 ….Block N)
For I=0 TO N
Rand_Number = Rand() % 4 // module of number (0,1,2,3)
Switch (Rand_Number)
{

```
Case 0: call Scenario_0(Block I); Break;
Case 1: call Scenario_1(Block I); Break;
Case 2: call Scenario_2(Block I); Break;
Case 3: call Scenario_3(Block I); Break;
}
Insert encrypted (Block I) to Stego message
Next I
Output Stego message

Routine Scenario_0(Block I) // Adding Kashida after pointed letters to encode one, otherwise, zero.
While Not_End_Block
Read letter
If letter connected then      // we can add Kashida
        {
                If (Hide bit==1) then
                Add Kashida
                Replace (Letter)
                Else
                No_change
        }
Next Letter
End while
End Routine Scenario_0
```

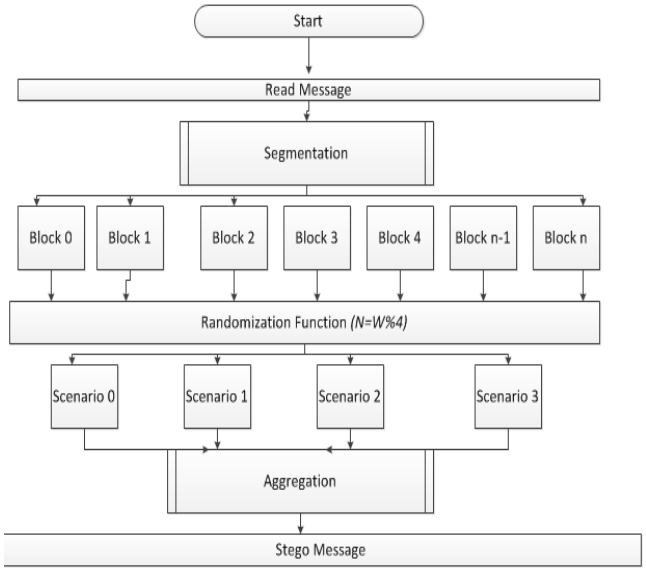

Figure 3. Message segmentation and aggregation

IV. ALGORITHM ANALYSIS AND ADVANTAGES

In our algorithm, we used variation and randomization to improve Kashida algorithm to hide data inside Unicode (Arabic language). In the following section, we explore and analytically discuss some of the advantages of the proposed algorithm.

Hiding capacity, transparency, and robustness represent the most important goals for any Steganography algorithm. In our algorithm, we achieve the following advantages:-By using the Kashida variation algorithm, we can pass more than one bit for a word. This is while other algorithms such as [9] can pass only one bit per word.

I. Our algorithm can be applied in different file formats like HTML files, and MS word files. This will also will reduce intruder suspicions.

II. Another advantage is that there is no need to create a database dictionary like [15] where searching for synonyms words will consume some time, and may give intruders a chance to analyze the transmitted files.

III. Randomization of our algorithm will improve robustness of hidden data, since in each round, the algorithm will pack up one of four states and this will prevent the intruder to figure out the algorithm procedure.

IV. Our algorithm can be extended to other similar Unicode languages like Urdu, Pashto, and Persian. Moreover our algorithm can be widely used, as Urdu and Arabic represent the 3rd and 4th top languages population [21].

V. By employing Kashida variation algorithm, unlike [9] the text semantic will not be changed.

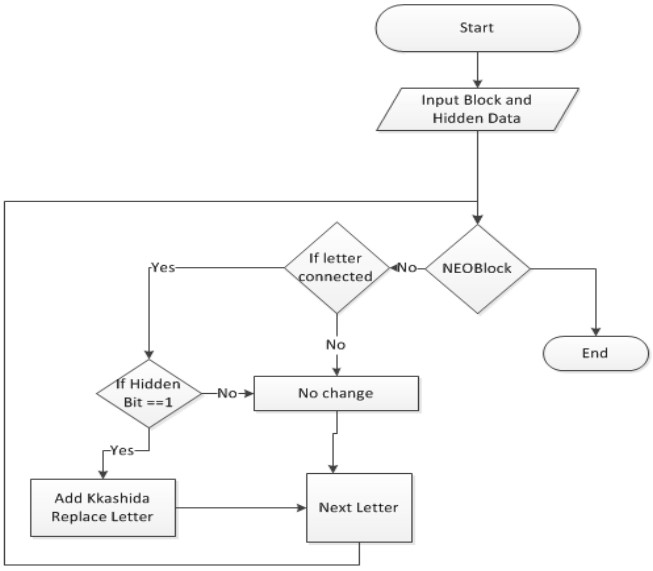

Figure 4.Routine Scenario_0 flow chart

V. EXPERIMENTS AND RESULTS

In this section, we explore some of the results acquired by applying our algorithm and considering the concept of hidden ratio.

Hidden ratio is defined as:

Hidden Ratio=amount of hidden data/carrier file size (1)

Table III shows the result of four scenarios suggested by the KVA algorithm and the position of data hidden inside it.

TABLE III. APPLIED SCENARIOS TO MEASURE HIDDEN RATIO

Applied Scenario	Hidden Bits	Stego Object
Scenario1	0100	اتق شر من احسنت اليه
Scenario2	1011	اتق شر من احسنت اليه
Scenario3	11000100	اتق شر من احسنت اليه
Scenario4	001011100	اتق شر من احسنت اليه

Table III shows an Arabic sentence "اتق شر من احسنت اليه" and the four suggested scenarios applied on it. Each scenario used Kashida (shift +J) to hide data depending on the position of the extension letter. So the semantic of the Stego-Object is the same and this avoids any suspicion from attackers. As shown in the Table, each scenario has a different hidden ratio, as adding Kashida characters depend on the inserted position. Hence the first two scenarios have less hidden data ratio compared to the last two. On average, the number of hidden data rate is six bits in the above message. So, depending on the experiments, inserting bits inside the message has three performance cases. The best case performance is as shown in scenario 3, while scenario 4 provides the worst case, and scenarios 1and 2 offer average case performance.

Table IV represents hidden data ratio and a comparison between our proposed algorithm and the word and line shifting algorithms.

TABLE IV. HIDDEN DATA RATIO

Web	Website	KVA	Line Shift	Word shift
1	www.aljazeera.net	34.5	0.20	2.87
2	www.bbc.co.uk	30.8	0.21	2.06
3	www.ahram.org.eg	35.5	0.23	2.96
4	www.addustour.com	41.2	0.32	2.75

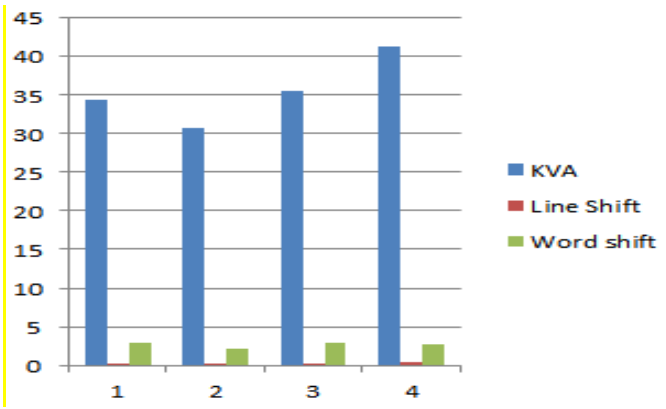

Figure 5. Hidden ratio

VI. CONCLUSION

Hiding data in different cover media represents one of the challenging security issues. One of the challenging media for hiding data is text, where embedding data may affect the text format, file size and format. File Changes will increase the probability of being discovered using Stegoanalysis tools and this will lead to revealing the hidden data. The strategy presented in this paper employed two concepts to improve transparency, robustness and hiding capacity by using the extension of Arabic language characters, called Kashida. Selection among different four scenarios increases the algorithm complexity, and also reduces the likelihood of being suspicions. In addition, this algorithm can be applied on different Unicode languages like Persian, Pashto, and Urdu, where nearly 2 billion of the world population can benefit from KVA.

REFERENCES

[1] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *IEEE computer,* vol. 31, pp. 26-34, 1998.

[2] A. Mangarae, "Steganography FAQ," *Zone-H.org,* 2006.

[3] S. Dickman, "An Overview of Steganography," *James Madison University Infosec Techreport, Department of Computer Science, JMU-INFOSEC-TR-2007-002 (http://www.infosec.jmu.edu/reports/jmu-infosec-tr-2007-002. pdf),* 2007.

[4] S. Changder, D. Ghosh, and N. Debnath, "Linguistic approach for text steganography through Indian text," in *Computer Technology and Development (ICCTD), 2010 2nd International Conference on*, 2010, pp. 318-322.

[5] T. Morkel, J. H. P. Eloff, and M. S. Olivier, "An overview of image steganography," in *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa*, 2005.

[6] M. A. F. Al-Husainy, "Image Steganography by mapping Pixels to letters," *Journal of Computer science,* vol. 5, pp. 33-38, 2009.

[7] V. Potdar and E. Chang, "Visibly Invisible: Ciphertext as a Steganographic Carrier," in *Proceedings of the 4th International Network Conference (INC2004)*, 2004, pp. 385-391.

[8] S. Bhattacharyya, I. Banerjee, and G. Sanyal, "A novel approach of secure text based steganography model using word mapping method (WMM)," *Journal on "International Journal of Computer and Information Engineering,* vol. 4, p. 2, 2010.

[9] R. Prasad and K. Alla, "A new approach to Telugu text steganography," in *IEEE Symposium on Wireless Technology and Applications (ISWTA)*, 2011, pp. 60-65.

[10] V. N. Rao and D. D. Shulman, *Classical Telugu poetry: an anthology*: University of California Press, 2002.

[11] K. Bennett, "Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text," *CERIAS Tech Report* p. 28, 2004.

[12] L. Yuling, S. Xingming, G. Can, and W. Hong, "An efficient linguistic steganography for Chinese text," in *IEEE International Conference on Multimedia and Expo*, 2007, pp. 2094-2097.

[13] S. H. Low, N. F. Maxemchuk, J. Brassil, and L. O'Gorman, "Document marking and identification using both line and word shifting," in Proceedings of the *Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People, IEEE INFOCOM'9.*, 1995, pp. 853-860.

[14] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM systems journal,* vol. 35, pp. 313-336, 1996.

[15] Z. H. Wang, C. C. Chang, and M. C. Li, "Emoticon-based text steganography in chat," in *Asia-Pacific Conference on Computational Intelligence and Industrial Applications, 2009. PACIIA,* 2009, pp. 457-460.

[16] M. L. Bensaad and M. B. Yagoubi, "High capacity diacritics-based method for information hiding in Arabic text," in *International Conference on Innovations in Information Technology (IIT)*, 2011, pp. 433-436.

[17] M. A. Aabed, S. M. Awaideh, A. R. M. Elshafei, and A. A. Gutub, "Arabic diacritics based steganography," in *IEEE International Conference on Signal Processing and Communications, ICSPC,* 2007, pp. 756-759.

[18] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "A new approach to Persian/Arabic text steganography," in *1st and 5th IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse Computer and Information Science, ICIS-COMSAR and IEEE/ACIS*, 2006, pp. 310-315.

[19] A. Odeh, A. Alzubi, Q. B. Hani, and K. Elleithy, "Steganography by multipoint Arabic letters," in *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2012, pp. 1-7.

[20] A. Azmi and A. Alsaiari, "Arabic Typography: A Survey," *International Journal of Electrical & Computer Sciences,* vol. 9, pp. 16-22, 2010.

[21] M. Turner, "The world's most widely spoken languages," ed: St. Ignatius High School. URL: http://www2.ignatius.edu/faculty/turner/languages. htm [viewed December 10, 2003], 2003.

Ammar Odeh is a PhD. Student in University of Bridgeport. He earned the M.S. degree in Computer Science College of King Abdullah II School for Information Technology (KASIT) at the University of Jordan in Dec. 2005 and the B.Sc. in Computer Science from the Hashemite University. He has worked as a Lab Supervisor in Philadelphia University (Jordan) and Lecturer in Philadelphia University for the ICDL Courses and as technical support for online examinations for two years. He served as a Lecturer at the IT, (ACS,CIS ,CS) Department of Philadelphia University in Jordan, and also worked at the Ministry of Higher Education (Oman, Sur College of Applied Science) for two years. Ammar joined the University of Bridgeport as a PhD student of Computer Science and Engineering in August 2011. His area of concentration is reverse software engineering, computer security, and wireless networks. Specifically, he is working on the enhancement of computer security for data transmission over wireless networks. He is also actively involved in academic community, outreach activities and student recruiting and advising.

Dr. Khaled Elleithy is the Associate Dean for Graduate Studies in the School of Engineering at the University of Bridgeport. He has research interests are in the areas of network security, mobile communications, and formal approaches for design and verification. He has published more than two hundred research papers in international journals and conferences in his areas of expertise. Dr. Elleithy is the co-chair of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE). CISSE is the first Engineering/Computing and Systems Research E-Conference in the world to be completely conducted online in real-time via the internet and was successfully running for six years. Dr. Elleithy is the editor or co-editor of 12 books published by Springer for advances on Innovations and Advanced Techniques in Systems, Computing Sciences and Software.

Dr. Miad Faezipour is an Assistant Professor in the Computer Science and Engineering program at the University of Bridgeport and the director of the Digital/Biomedical Embedded Systems & Technology (D-BEST) Lab since July 2011. Prior to joining UB, she has been a Post-Doctoral Research Associate at the University of Texas at Dallas collaborating with the Center for Integrated Circuits and Systems and the Quality of Life Technology laboratories. She received the B.Sc. in Electrical Engineering from the University of Tehran, Tehran, Iran and the M.Sc. and Ph.D. in Electrical Engineering from the University of Texas at Dallas. Her research interests lie in the broad area of biomedical signal processing and behavior analysis techniques, high-speed packet processing architectures, and digital/embedded systems. Dr. Faezipour is a member of IEEE, EMBS and IEEE women in engineering.