STEGANOGRAPHY IN ARABIC TEXT USING ZERO WIDTH AND KASHIDHA LETTERS

Ammar Odeh[1]and Khaled Elleithy[2]

[1]Department of Computer Science & Engineering, University of Bridgeport
Bridgeport, CT06604, USA
aodeh@bridgeport.edu
[2]Department of Computer Science & Engineering, University of Bridgeport
Bridgeport, CT06604, USA
elleithy@bridgeport.edu

ABSTRACT

The need for secure communication methods 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 Arabic text. The algorithm employs some letters that can be joined with other letters. These letters are the extension letter, Kashida and Zero width character. The extension letter, Kashida, does not have any change in the word meaning if joined to other letters. Also, the Zero width character (Ctrl+ Shift +1) does not change the meaning. The new proposed algorithm, Zero Width and Kashidha Letters (ZKS), mitigate the possibility to be discovered by steganoanalysis through using parallel connection and permutation function.

KEYWORDS

Steganography, Kashida, Carrier file, Zero width character, textsteganography, image steganography, audio steganography, Information Hiding, Persian/Arabic Text, Stegoanalysis,stego_medium, stego_key.

1. INTRODUCTION

1.1.BACKGROUND

Steganography is a Greek word coming from cover text. "Stegano" means hidden and “Graptos" means writing. In steganography, the secure data will be embedded into another object, so middle attacker cannot catch it [1]. Invisible ink is an example for Steganography using a readable message transfer between source and destination. Everyone in the middle can read the message without having any clue about the hidden data. On other hand, authorized persons can read it depending on the substances features [2][3].

Ancient Greeks used to shave the messenger head and then wait until the hair grew back. That is when the message will be sent to the destination [1]. Depending on this method, there are two possibilities:

1. Message has arrived so the receiver can read the message and recognize if message has changed or not.
2. If message did not arrive, it means the attacker has detected the message.

DOI : 10.5121/ijcsit.2012.4301

1.2. MOTIVATION

Steganography algorithms depend on three techniques to embed the hidden data in the carrier files.

1. Substitution: Exchange a small part of the carrier file by the hidden message where the middle attacker cannot observe the changes on the carrier file. On the other hand, in choosing a replacement process, it is very important to avoid any suspicion. This means that it is important to select insignificant parts from the carrier file and then replace them. For instance, if the carrier file is an image (RGB), then the least significant bit (LSB) can be used as the exchange bit [4].

2. Injection: By adding hidden data into the carrier file, the file size will increase and this will increase the suspicion. Therefore, the main goal to present techniques to add hidden data while avoiding attacker suspicion [4].

3. Propagation: There is no need for a cover object. It depends on using a generation engine fed by input (hidden data) to produce and mimic a file (graphic or music or text document).

The Steganography process consists of three main components as show in Figure 1.

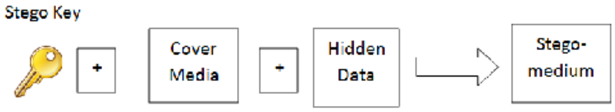


Figure 1. General components of Steganography

Different types of cover media including image, sound, video and text can be used in Steganograph, as shown in Figure 2. Choosing carrier file is very sensitive where it plays a key role to protect the embedded message. Successful Steganography depends on avoiding suspicion. Steganalysis will start checking the file. If there is any suspicion, this will compromise the main goal of Steganography [3][4].

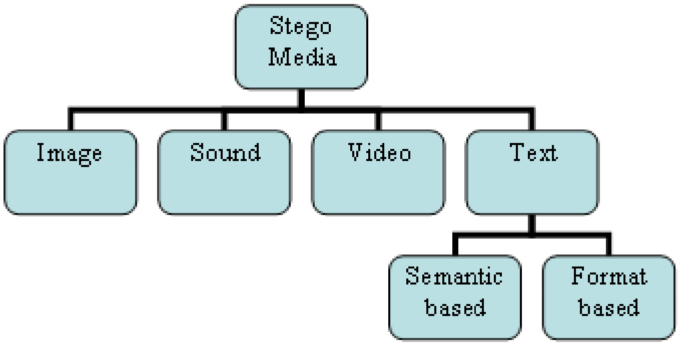


Figure 2. Stego Media

Text Steganography represents the most difficult type, where there is generally lack of data redundancy in the text file in comparison with other carrier files [5]. The existence of such redundancy can help increase the capacity of hidden data size. Furthermore, text Steganography depends on the language, as each language has its own unique characteristics, which is completely different from other languages. For example, the letter shape in English language does not depend on its position in the word, while Persian/Arabic letters have different forms depending on letter positioning [6].

In our new proposed algorithm, we hide text inside text by employing Arabic language and applying a random algorithm to distribute the hidden bits inside the message. The main reasons for choosing the Arabic language are:

1. The proposed algorithm will depends on multi dotted points letters. Therefore, the algorithm must employ a language that has as many as possible dotted letters. For example, the Arabic Language has 5 multipoint letters and Persian/Farsi language has 8 letters [7], while English does not have any.
2. Wealth availability of electronic textual information.
3. There is little research on other languages compared to English.
4. The approach can be extended to other languages like Urdu and Kurdish.

1.3.MAIN CONTRIBUTION AND PAPER ORGANIZATION

An efficient algorithm is presented in this paper. The main idea is to use Kashida, Zero width characters in Arabic that enables us to hide more tow bits per one letter, Most of pervious algorithm hide one bit for one letter. Addition we will use parallel connection, randomization strategy to avoid any adaption.

The rest of this paper is organized as follows. In section II we discuss some text Steganography techniques. Employ Kashida and Zero width hidden algorithm discuses in section III. Finally, conclusion remarks are in section IV.

2. PRIOR WORK

Text Steganography is divided into two categories. The first one is the semantic method, and the second is the formatting method, as shown in Figure 2. In this Section, we will briefly explain some Steganography examples. In Table I, we present a simple comparison between semantic and formatting methods.

Table I. Comparison between text Steganography methods

	Semantic Method	Format Method
Amount of hidden data	Small amount	More than semantic
Flaws	Sentence meaning	notice from OCR or retyping

Steganography criteria will depend on the amount of data that can be hidden and the main problem facing the method.

We describe ten algorithms that hide data inside text documents. The last two algorithms deal with Arabic and Persian languages.

2.1. WORD SYNONYM

Word Synonym is also called semantic method and it depends on replacing some words by their synonym. See Table II. This technique will convey data without making any suspicion. It is limited in terms of that fact that hidden data will be small relative to other methods. Moreover, it may change the sentence meaning [7][10][12].

2.2. PUNCTUATION

This method uses punctuation like (.)(;) to represent hidden text. For example "NY, CT, and NJ" is similar to "NY, CT and NJ" where the comma before the “and” represents 1, and the other represents 0. The amount of hidden data in this method is very small compared to the amount of cover media. Inconsistence use of punctuation will be noticeable from Stegoanalysis point of view [9].

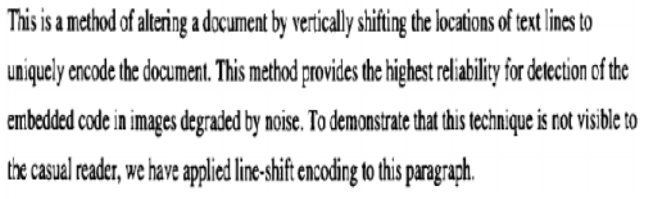
Table II. Using Word Synonym

Word	Synonym
Big	Large
Find	Observe
Familiar	Popular
Dissertation	Thesis
Chilly	Cool

2.3. LINE SHIFTING

Line shifting means to vertically shift the line a little bit to hide information to create a unique shape of the text. Unfortunately, line shifting can be detected by a character recognition program. Moreover retyping removes all hidden data [7][10].

In Figure 3, we present an example regarding line shifting where the vertical shifting is very small (1/300 inch). This is not noticeable by the human eye.



This is a method of altering a document by vertically shifting the locations of text lines to
uniquely encode the document. This method provides the highest reliability for detection of the
embedded code in images degraded by noise. To demonstrate that this technique is not visible to
the casual reader, we have applied line-shift encoding to this paragraph.

Figure 3. Line shifting; second line is shifted up 1/300 inch [10].

2.4. WORD SHIFTING

In this method, changing spaces between words enables us to hide information. Word shifting is noticeable by OCR through detecting space sequence between words [7][10].

2.5. SMS ABBREVIATIONS

Recently most SMS messages use abbreviations for simplicity and security while used in different applications such as internet chatting, email, and mobile messaging. The main advantage of this method is to speed typing, reducing the message’s length and manipulated keyboard limitation character [13].

Other algorithms use numbers to convey specific information. As mentioned above, SMS abbreviation can be used in specific applications while using in others creates suspicion of any entity that monitors the ongoing transmission.

2.6. TEXT ABBREVIATIONS

Text abbreviation is similar to SMS abbreviation, where a dictionary is created for each word abbreviation and its meaning. The dictionary is published between the communication parties. Abbreviation represents one method to hide data. For example if you send (see) it means (do you understand) [13].

Table III. Some SMS Abbreviations

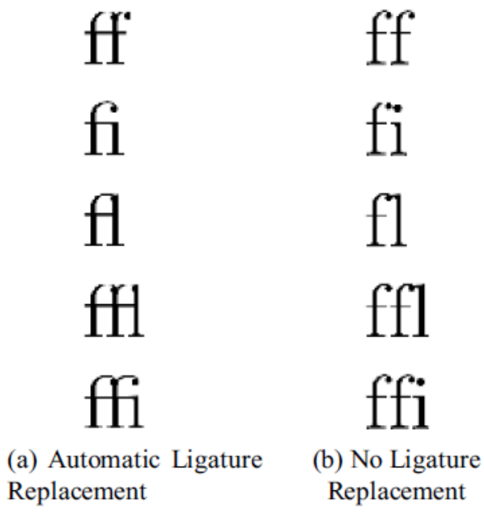
Abbreviation	Meaning
ADR	Address
ABT	About
URW	You are welcome
ILY	I love you
EOL	End of lecture
AYS	Are you serious?

2.7. HTML SPAM TEXT

This method depends on HTML pages, where their tags and their members are insensitive. For example <BR> equal to <Br>, and the same as <br> and <bR>. The hidden data depends on upper case or lower case letters to embed0 or 1.

2.8. TEX LIGATURES

In TeX ligatures, some special groups of letters can be joined together to create a single glyph as shown in Figure 4. The algorithm finds available ligatures in the text to hide a single bit in each one. For example, if we want to hide 1 we write fi to f {} i which creates some space between f and i. Otherwise, we encode 0 [5].



(a) Automatic Ligature Replacement (b) No Ligature Replacement

Figure 4 .Join between characters [5]

The same algorithm can be applied to Arabic character "لا" or "لا". This algorithm has two problems. The first problem is that file size increases when we apply extension in our text. The second problem is that if the ORC notices the font change, it can detect the decoding hidden message [6][5].

2.9. ARABIC DIACRITICS

Arabic language uses different marks. The main reason to use these symbols is to distinguish between words that have same letters. It depends on Arabic Diacritics (Harakat), where diacritics are optional. Most of Arabic novels can be read without Diacritics, which depends on the language's grammar. The most occurrence is Fatha " ◌َ" which will be used to encode 1 otherwise encode 0.Our new algorithm will enhance the reuse of cover media. Furthermore, the carrier file size might be reduced depending on the hidden message. On the other hand, when ORC detects the same message with different diacritics, it might conclude that there is a hidden data. In addition, retyping will remove the embedded message [8].

Table IV. Some Letters with mark and their Pronunciation

Haraka	Letter with Haraka	Pronunciation
Dama	دُ	Do
Kasra	دِ	De
Fatha	دَ	Da

2.10. VERTICAL DISPLACEMENT OF THE POINTS

This algorithm achieves excellent performance as it is applied on pointed (dotted) letters. Other languages such as English language have only two dotted letters; {i, j}; and thus limits the application of this algorithm. Alternatively, some languages such as Arabic and Persian have many pointed letters which make them fit better for this technique.

Arabic and Persian languages have many pointed characters. Arabic has 26 letters where 13 of them are pointed, and Persian has 32 letters where 22 of them are pointed. In this new algorithm, we encode 1 to shift up the point, otherwise encode 0. This method can encode a huge number of bits, and need a strong OCR to recognize the changes. Meanwhile, retyping will remove the entire message [7].

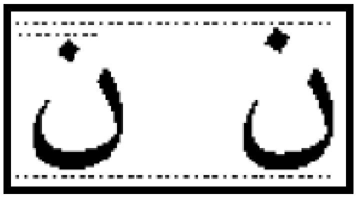
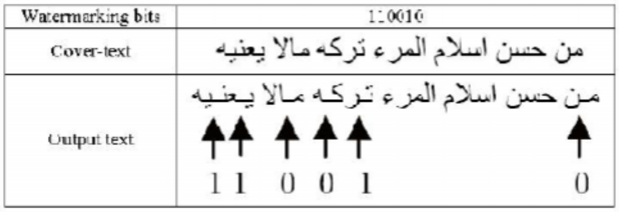


Figure5. Vertical shifting point [7]

2.11. USING THE EXTENSION 'KASHIDA' CHARACTER

Strategy of this method will depend on letter extension (Kashida). Kashida cannot be adding at the beginning and at end of word, it can be added between letters in words. In other words if un-pointed letter with extension to hide zero, pointed letter with extension will hold one.

Message content will not be affected. On other hand, a new Unicode will be added (0640).



Watermarking bits	110010
Cover-text	من حسن اسلام المرء تركه مالا يعنيه
Output text	مـن حسن اسلام المرء تـركـه مـالا يـعـنـيه 1 1 0 0 1 0

Figure6. Vertical shifting point [6]

As Figure show, not all characters will hide a bit. Therefore, stegoanalysis may suspect message and this will vaulted main goal of steganography.[13]

2.12. UTILIZATION OF USING THE EXTENSION 'KASHIDA' CHARACTER

This algorithm try to use Kashida by the following way, One Kashida represent 0 and 2 Kashida represent one. More over depending on number of Arabic letters with sum up is 32. Since each letter, need 16 bits to represented, the algorithm use-mapping table to which each character map. So instead 16bit we can represent each letter by 6 bit only and this will save 10 bits.[14]

2.13. USING PSEUDO-SPACE AND PSEUDO CONNECTION CHARACTERS

Also called zero width non-joins (ZWNJ) and zero width joiner (ZWJ) characters. At the beginning, we classify letters to join or non-join letters.If we want to hide 1 we will add zero width, otherwise we hide 0.[15]

3. PROPOSED ALGORITHM

Some of Arabic characters features support different steganography algorithms.

3.1. ARABIC LETTERS CHARACTERISTICS

A. Arabic language has 28 letters and each one of them has four different shapes, depends on position of that letter. English language letter have same shape regardless position. Table V show some Arabic letter shape

Table V. Some Letters with mark and their Pronunciation

Letter	beginning	Middle	End	
ت	ت	ـتـ	ـت	*b*
ث	ﺛ	ـثـ	ـث	*t*

B. Most of Arabic letters can be connected together like (يعلمون) where in English all words consist of separated letters.

C. Each Arabic letters encoding into Unicode. Where each letter represents by 2bytes.

D. Any steganography algorithm applied in Arabic text can be extended to other language. (Persian, Pashto, Sindhi, Kurdish, Urdu).

3.2. ZKS ALGORITHM

ZKS algorithm tries to employ letters connectivity and extension to hide 1 bit, moreover using Zero width letter to hide 2 bits per each connective character.

Table VI: - Steganography Extension algorithm

Cover Object	كان ساحل مصر الشمالي سلّة غذاء مصر والامبراطورية الرومانية التي كانت تحتل مصر قبل الإسلام. فقبل بناء السد العالي، اعتمد المصريون على مياه النيل فى الزراعات الصيفية في الوادي والدلتا كما اعتمدوا على مياه الأمطار في زراعة القمح
Stego Object	كـان ساحـل مـصـر الشمـالي سـلـّة غذاء مـصـر والامبراطـورية الرومـانـيـة الـتي كانت تـحـتـل مـصـر قـبل الإسـلـام. فقبل بـناء السـد العـالي، اعتمد الـمصريـون عـلـى مـيـاه الـنـيل فـى الزراعـات الصـيـفـيـة في الـوادي والدلتا كـمـا اعـتـمـدوا عـلـى مـياه الأمطـار في زراعة القـمـح

Hidden Bits	110101010110111001001111001111111110110101
	110110001001111111010001111111001000111 1
	11100100011

As table above show, a huge amount of bits can be added to message. In by applying this algorithm we can hide a huge amount of data.

By applying Zero width letter (U+200D) we can increase hidden bit capacity (Ctrl+Shift+1).

Table VII: - Steganography Extension algorithm

Extension	Zero Width	Code	Letter effect
No	No	00	No EFFECT
Yes	No	01	Extension
No	Yes	10	Zero width
Yes	Yes	11	Extension + Width

Table VIII: - Simulated results by applying ZKS algorithm

Cover Object	كان ساحل مصر الشمالي سلّة غذاء مصر والامبراطورية الرومانية التي كانت تحتل مصر قبل الإسلام. فقبل بناء السد العالي، اعتمد المصريون على مياه النيل فى الزراعات الصيفية في الوادي والدلتا كما اعتمدوا على مياه الأمطار في زراعة القمح
Stego Object	كـان ساحـل مـصـر الشمـالي سـلـّة غذاء مـصـر والامبراطـورية الرومـانـيـة الـتي كانت تـحـتـل مـصـر قـبل الإسـلـام. فقبل بـناء السـد العـالي، اعتمد الـمصريـون عـلـى مـيـاه الـنـيل فـى الزراعـات الصـيـفـيـة في الـوادي والدلتا كـمـا اعـتـمـدوا عـلـى مـياه الأمطـار في زراعة القـمـح

Hidden Bits	100100100000111010110001011110110001111 1
	110011101111100111110100000100101110001 1
	0000000101101000111111000000101001

As the above table VIII shown the mount of bits can be hidden inside text message. The amount of change in text unnoticeable.

3.3. PSEUDO CODE AND FLOW CHART

ZKS algorithm uses two stages to hide message, to avoid any attacker suspicions.

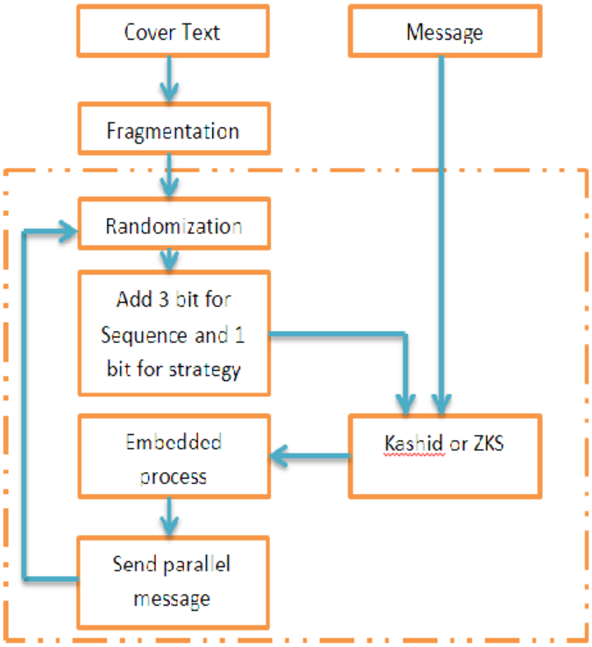


Figure7. State Diagram for ZKS

The first stage depends on Fragment of Stego cover media to enable different strategies to apply to embedded message.

In the figure, we suggest eight messages can be send parallel, so we send embedded 3 bits to recognize sequence number of that message. Depends on message to be hiding we can add bit to increase of parallel messages as show in figure 8.

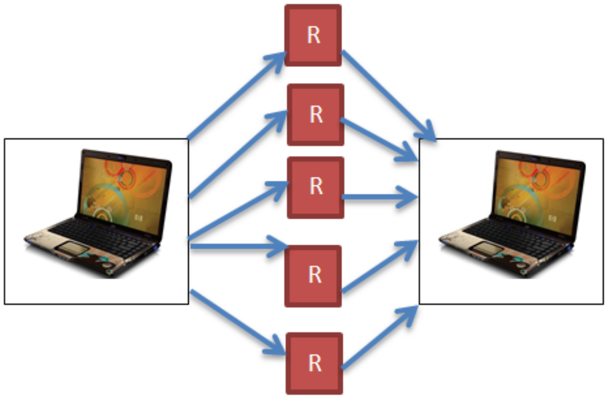


Figure8. Parallel connection

In second stage, algorithm permutated-fragmented messages and randomization function choose which application can be used.

Therefore, the first four most significant bits determine sequence message and last one for applied algorithm.

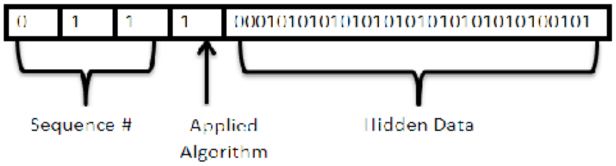


Figure9. State Diagram for ZKS

Each message has different Stego key regardless of routing path, which increase Stegoanalysis confusion.

3. Conclusion

In our paper, we introduced new text Steganography in Arabic letters. Our algorithm deals with connected letter by adding Kashida character and Zero width letter. ZKS algorithm improve previous one by use different concepts like parallel connection, permutation, and randomization, to complicated Stegoanalysis process.

References

[1] Aelphaeis Mangarae "Steganography FAQ," Zone-H.Org March 18th 2006

[2] S. Dickman, "An Overview of Steganography," July 2007.

[3] V. Potdar, E. Chang. "Visibly Invisible: Ciphertext as a Steganographic Carrier," Proceedings of the 4th International Network Conference (INC2004), page(s):385–391, Plymouth, U.K., July 6–9, 2004

[4] M. Al-Husainy "Image Steganography by Mapping Pixels to Letters," 2009 Science Publications

[5] M. Shahreza, S. Shahreza, "Steganography in TeX Documents," Proceedings of Intelligent System and Knowledge Engineering, ISKE 2008. 3rd International Conference, Nov. 2008

[6] M. S. Shahreza, M. H. Shahreza, "An Improved Version of Persian/Arabic Text Steganography Using "La" Word" Proceedings of IEEE 2008 6th National Conference on Telecommunication Technologies.

[7] M. H. Shahreza, M. S. Shahreza, "A New Approach to Persian/Arabic Text Steganography " Proceedings of 5th IEEE/ACIS International Conference on Computer and Information Science 2006

[8] M. Aabed, S. Awaideh, A. Elshafei and A. Gutub "ARABIC DIACRITICS BASED STEGANOGRAPHY" Proceedings of IEEE International Conference on Signal Processing and Communications (ICSPC 2007)

[9] W. Bender ,D. Gruhl ,N. Morimoto ,A. Lu "Techniques for data Hiding" Proceedings OF IBM SYSTEMS JOURNAL, VOL 35, NOS 3&4, 1996

[10] K. Bennett, "Linguistic Steganography : survey, analysis, and robustness concerns for hiding information in text" Center for Education and Research in Information Assurance and Security, Purdue University 2004

[11] M. Nosrati , R. Karimi and, M. Hariri ," An introduction to steganography methods" World Applied Programming, Vol (1), No (3), August 2011. 191-195.

[12] M.H. Shirali-Shahreza, M. Shirali-Shahreza, " Text Steganography in chat" Proceedings of 3rd IEEE/IFIP International Conference in Central Asia on Sept. 2007

[13] Adnan Abdul-Aziz Gutub, Wael Al-Alwani, and Abdulelah Bin Mahfoodh" Improved Method of Arabic Text SteganographyUsing the Extension „Kashida" Character" Bahria University Journal of Information & Communication Technology Vol.3, Issue 1, December 2010

[14] Adnan Abdul-Aziz Gutub, and Manal Mohammad Fattani, “A Novel Arabic Text Steganography Method Using Letter Points and Extensions “ World Academy of Science, Engineering and Technology 27 200

[15] Hassan Shirali-Shahreza, Mohammad Shirali-Shahreza "STEGANOGRAPHY IN PERSIAN AND ARABIC UNICODE TEXTS USING PSEUDO-SPACE AND PSEUDO CONNECTION CHARACTERS". Journal of Theoretical and Applied Information Technology

Authors

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 th e 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. 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 one hundred fifty 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 four years. Dr. Elleithy is the editor or co-editor of 10 books published by Springer for advances on Innovations and Advanced Techniques in Systems, Computing Sciences and Software.