Steganography by Multipoint Arabic Letters

Ammar Odeh, Aladdin Alzubi, Qassim Bani Hani, Khaled Elleithy
Department of Computer Science & Engineering,
University of Bridgeport
Bridgeport, CT 06604, USA
aodeh @bridgeport.edu

***Abstract*-Security methodologies are taken into consideration for many applications, where transferring sensitive data over network must be protected from any intermediate attacker. Privacy of data can be granted using encryption, by changing transmitted data into cipher form. Apart from encryption, hiding data represents another technique to transfer data without being noticeable by an attacker. This technique is called Steganography. In this paper, we will discuss the main concepts of Steganography and a carrier media that is used for this goal. Employing text as mask for other text represents the most difficult method that can be used to hide data. We will discuss some algorithms that use Arabic text. We then describe our doted space methodology to enhance data hiding.**

Keywords: Steganography, carrier file, text steganograph, image steganography, audio steganography, information hiding, Persian/Arabic Text, steganalysis, stego medium, stego_key.

I. INTRODUCTION

A. 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 can't 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.

B. 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. So 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.

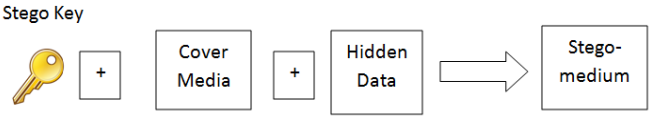


Figure 1. General components of Steganography

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].

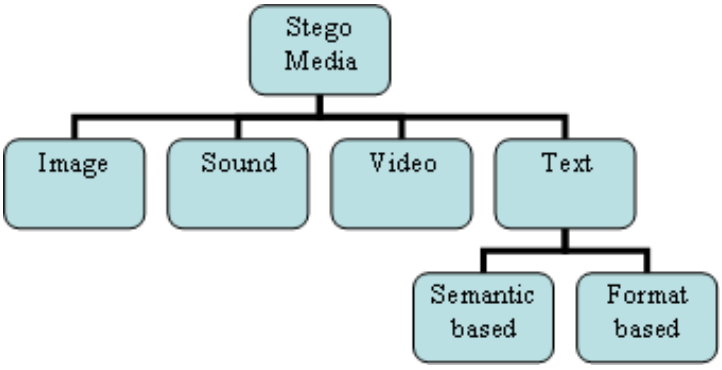


Figure 2. Stego Media

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.

C. Main Contribution and Paper Organization

An efficient algorithm is presented in this paper. The main idea is to use multiple point characters in Arabic which enables us to hide more than two bits per letter. The rationale behind this approach is that most of the previous algorithms reported in literature hide one bit for one letter. Furthermore, we will merge our algorithm with vertical shifting point algorithm to increase the size of the hidden file. The size of the carrier will be constant without any change. After we add the data, we convert the file into image to avoid the retyping problem.

The rest of this paper is organized as follows. In section II, we discuss some text Steganography techniques as well as their advantages and disadvantages. Our multipoint hidden algorithm is discussed in section III. In section IV, we present experimental results of our algorithm. Finally, conclusions are presented in section V.

II. 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.

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. 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 Steganoanalysis point of view [9].

Table II. Using Word Synonym

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

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.

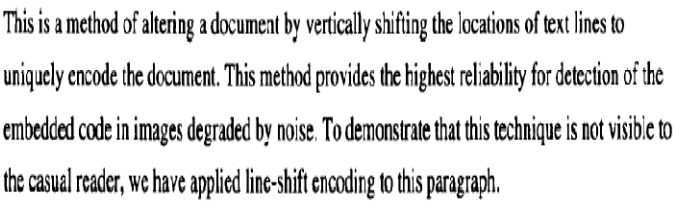


Figure 3. Line shifting; second line is shifted up 1/300 inch [10].

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].

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.

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?

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].

7. HTML Spam Text

This method depends on HTML pages, where their tags and their members are insensitive. For example
 equal to
, and the same as
 and
. The hidden data depends on upper case or lower case letters to embedd 0 or 1.

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].

The same algorithm can be applied to Arabic character "ﻉ" or "ﺵ". 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].

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].

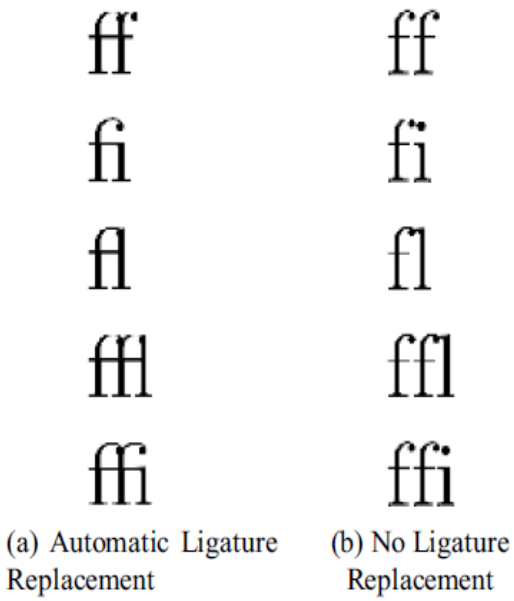


Figure 4 .Join between characters [5]

Table IV. Some Letters with mark and their Pronunciation

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

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]

III.PROPOSED ALGORITHM

Pointed letters represents one of the important characteristics of Arabic and Persian languages. Table II classifies Arabic letter with respect to the number of points.

Table V Arabic letters with respect to the number of points

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

Our algorithm hides data in multipoint Arabic/Persian letters like (ث, tha). In Arabic language, there are five multi-pointed letters, and in Persian there are eight. Each character can be used to hide 2 bits to determine the shifting and distance between letter points. Table III represents the relation between letter, shifting, and encoding

Table VI represents relation of shifting and distance to letter format

Point shifting	distance	code	Letter effects
0	0	00	No change
0	1	01	Only distance between point increase
1	0	10	Only little upper shifting
1	1	11	Upper shifting and increase distance between point

A. Pseudo Code and Flow Chart

In this subsection we present the pseudo-code and flowchart of the proposed Arabic multipoint steganography algorithm. The flowchart of the algorithm is shown in Figure 7. The pseudo-code follows:

1. Enter the text and hidden file and its size
2. Search for multipoint letters
3. Hide size of embedded data at the beginning
4. For I= start to EOF
   IF hidden data ="00" then call Nochange();
   Else if hidden data= "01" then call distance ();
   Else if hiddendata ="10" then call shifting ();
   Else if hiddendata="11" then call distance_shifting();
   Else random call for any one // for padding purpose
   End for
5. Convert file to image file and send to other side
6. End

As can be seen in the above pseudo-code, in step 5, the file must be converting to image file for transfer. The receiver will scan the image file and find out the multipoint letters and then classify the function applied on it.

B.Advantages and Disadvantages

Our multipoint algorithm has many advantages, as one character can hide 2 bits compared to other algorithms that can hide 1 bit per letter. This implies that the amount of hidden data can be duplicated. Furthermore, the number of changed characters for a given message, which leads to decrease which leads to less suspicion. Moreover, the file size will have fewer changes, since the number of changes in characters format is less.

On the other hand, any retyping process removes all the hidden data, as the hidden data depends on the file format. The consistent format used in the system might raise the level of suspicion of an attacker.

IV. EXPERIMENTAL RESULTS

A.Multipoint algorithm

Our algorithm depends on multipoint letters to include hidden data. For this reason, we test different websites that contain text and picture.

We will run two files in parallel for carrier file and hidden data file. This process we will use two bits for each letter by applying the *distance_shifting* algorithm.

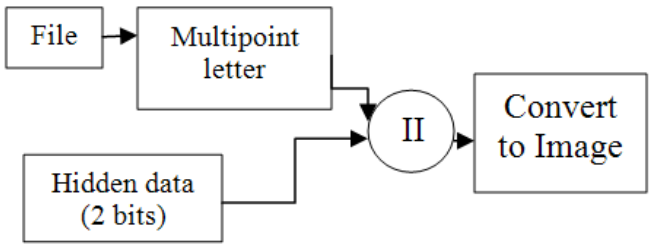


Figure 6. Parallel operation carrier file and hidden data

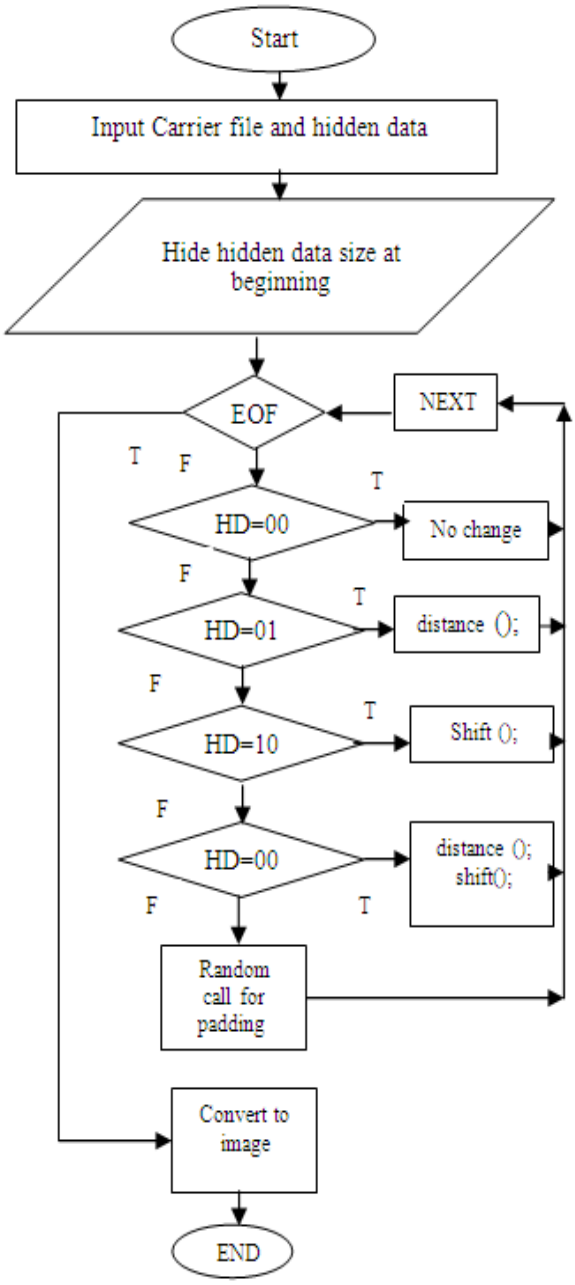
As shown in Figure 6, after we merge the hidden data, we convert it to image file to prevent the issues caused by retyping. After that, we can compress the file to produce the compressed image. The receiver decompresses the image and then extracts the hidden data.

B. Experimental Results of the Multipoint Algorithm

Table VII Capacity of webpage for different Arabic website

#	Page Name	Page Size	Character # 2 point or more	Capacity Ratio (Bit/ Kilobyte)
1	aljazeera.net	23.8 KB	1245	105
2	daralhayat.com	15.4 KB	968	126
3	salahws.com	10.3 KB	535	104
4	holyquran.net/tadabur	13.8 KB	516	75
5	khayma.com	21.8	499	46

		KB		



HD: Hidden Data
EOF: End of File

Figure 7. Flowchart of multipoint algorithm

As shown in Table VII, we used different Arabic news websites to observe the data ratio that can be hidden in each website.

C. Analysis of the Algorithm

Our algorithm is compared to the vertical shifting algorithm in terms of the number of changing letters and the number of bits that can be hidden. In Table VIII, the result of testing one paragraph is shown. The total number of letters is 115 which indicates that 42% of it is pointed. On the other hand, the number of multipoint letters is 29, which indicates 25%. The data that can be hidden in the multiple point algorithm is more than the first one. So we calculate the efficiency depending on Equation 1:

$$E = \#of\ hidden\ characters / \#of\ characters \quad *100\% \qquad (1)$$

where E is the efficiency.

So, the efficiency of multiple points is 50% while vertical point shifting is 43%.

Table VIII Vertical Point shifting versus Multipoint algorithm

أثبتت دراسة حديثة أجراها مجموعة من الباحثين البريطانيين أنه يمكن التعرف على مفاتيح شخصية بعض الأشخاص من خلال التدقيق في نوع سيارته الخاصة[13]		
	Number of letter	Number of hidden bit
Pointed letter	50	50
Multipoint letter	29	58

D. Merge with Vertical Point Shifting

In this subsection, we calculate the ratio when we compound the vertical point shifting algorithm discussed in [7] with the multipoint algorithm.

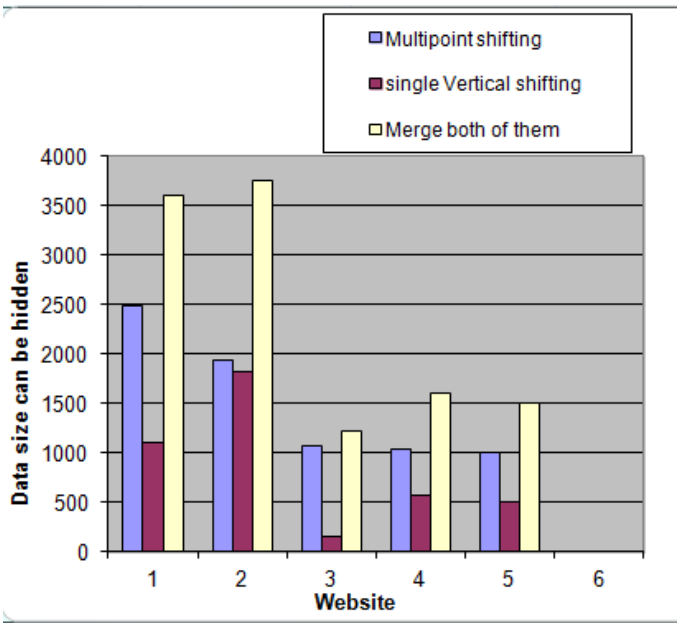


Figure 8. Merging vertical and multipoint algorithms

In Figure 8, we note the ability of the vertical point algorithm to hide huge amount of data as compared to the multipoint algorithm. On the other hand, an observer can notice the vertical changes more than multipoint changes where the number of changes will be more. Consequently, the

merging both algorithms gives us better idea and flexibility in hiding data and those that can be detected by an observer.

V. CONCLUSION

In this paper we introduce a new text Steganograph for Arabic multipoint letters. The new algorithm deals with two bits for each multipoint letter. We combine our strategy with vertical point shifting [7] to improve the amount of hidden data.

The retyping process is a challenging problem for similar algorithms which removes all the hidden data. We solve this challenge to mitigate any new font format changes by unifying all data which leads a homogenous file. Finally, the result reported by this implementation has outperformed similar results reported in literature in terms of the hiding capacity and the possibility of having such steganography mechanism used in hiding information.

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), pp. 385–391, Plymouth, U.K.*, July 2004.

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

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

[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 Steganograophy" *Proceedings of IEEE International Conference on Signal Processing and Communications (ICSPC'07)*, 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] J. Brassil, , S. Low, N. Maxemchuk, L. Gorman, "Electronic Marking and Identification Techniques to Discourage Document Copying", in *Proceedings of the 13th IEEE INFOCOM Networking for Global Communications Conference,* Oct.1995.

[11] 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.

[12] M. Nosrati , R. Karimi and, M. Hariri ," An Introduction to Steganography Methods" *World Applied Programming, Vol. 1, No. 3,pp. 191-195,* Aug. 2011.

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

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.

Qassim Bani Hani is Ph.D. candidate of computer science and Engineering department in the University of Bridgeport. His current research interests include the design and development of learning environment to support the learning about heterogamous domain, collaborative discovery learning and the development of mobile applications to support mobile collaborative learning (MCL), The congestion mechanism of transmission of control protocol including various existing variants, delivery of multimedia applications. He completed his Bachelor degree in computer science from Irbid National University in 2004 and Master degree in computer science from Al-Balqa' Applied University in 2007. Qassim has been directly involved in design and development of mobile applications to support learning environments to meet pedagogical needs of schools, colleges, universities and various organizations.

Aladdin Alzubi received the B.Sc. in Software Engineering from Philadelphia University, Amman, Jordan in 2004, and the Master of Computer Sciences from University Sians Malaysia – Malaysia in 2006. In 2011 he joined University of Bridgeport as Ph.D. student in computer science and engineering at the University of Bridgeport, Connecticut-USA. From 2000 to 2004.

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.