# A New Forward Secure Elliptic Curve Signcryption Key Management (FS-ECSKM) Scheme for Heterogeneous Wireless Sensor Networks

[1]Dr. Esam A. A. A. Hagras, [2]Doaa El-Saied, [3]Dr. Hazem H. Aly
[1]Alexandria University, Faculty of Eng., Electrical Department, Alexandria, Egypt
[2,3]Arab Academy of Science & Technology and Maritime Transport (AAST), Fac., of Eng, Cairo, Egypt

**Abstract**
In this paper, an efficient Forward Secure Elliptic Curve Signcryption Key Management (FS-ECSKM) Scheme for Heterogeneous Wireless Sensor Networks (HWSN) has been proposed. The proposed protocol is optimized for cluster sensor networks and is efficient in terms of complexity, number of message exchange, computation, and storage requirements with optimized security benefits for clustered environment. In addition to the message confidentiality,authentication, unforgeability and non repudiation, the proposed scheme achieves forward secrecy, public verification, and encrypted message authentication. The performance evaluation shows that this scheme can provide also extra saving in storage space compared with other previous schemes.

**Keywords**
Heterogeneous sensor networks, Elliptic curve cryptography, Signcryption, Forward secrecy.

## I. Introduction

Providing security for wireless sensor networks (WSN) is a challenging task. To overcome such challenges, many security primitives are required such as data authentication, unforgeability, non repudiation, and confidentiality of message contents; however these primitives require the usage of keys for encryption/ decryption/signature validation. These keys need to be transported and managed in a secure and efficient manner. However, WSN by their resource constrained nature, present many challenges for key transport and management. The key transmission and management protocols should not rely on heavy processing due to the constrained processing and battery life of the sensors. Also the utilized protocols should be very flexible in operation and does not depend on fixed topology due to the dynamic nature of the network and should accommodate the fact that many nodes will enter or exit the network at any time. Moreover, it should be scalable so that network size should not impact the network operation significantly.

Finally, key management scheme should be robust for node compromise and could be easily recovered. This compromise should not compromise the entire network, and should be constrained to a section of the network. Previous research on sensor network security mainly considers homogeneous sensor networks, where all sensor nodes have the same capabilities. Research has shown that homogeneous ad hoc networks have poor performance and scalability. Several recent works studied HWSNs, where sensor nodes have different capabilities in terms of communication, computation, energy supply, storage space, reliability and other aspects [1,2].

Sensor networks must arrange several types of data packets, including packets of routing protocols and packets of key management protocols. The key establishment technique employed in a given sensor network should meet several requirements to be efficient. These requirements may include supporting in-network processing and facilitating self-organization of data, among others. However, the key establishment technique for a secure application must minimally incorporate authenticity, confidentiality, integrity, scalability, and flexibility.

Xiaojiang Du and Hsiao Chen proposed a novel routing-driven key management scheme based on Elliptic Curve Cryptography (ECC) for HWSNs [1]. This scheme achieved better security with significant reductions on communication overhead, storage space, and energy consumption than other previous key management schemes. Also we have designed an efficient routing-driven key management protocol based on public key elliptic curve signcryption scheme for HWSNs [2]. This protocol has been optimized for cluster sensor networks and was efficient in terms of complexity, number of message exchange, computation, and storage requirements with optimized security benefits for clustered environment. Also it achieved significant reductions on storage space, and energy consumption compared with [1]. But both schemes lacked forward secrecy, public verification, and encrypted message authentication.

In this paper, we combine signcryption and forward secure [3-7] to present a new key management scheme based on elliptic curve signcryption with forward secrecy for HWSNs. This scheme not only provides forward secrecy but also achieve public verifiability. By forward secrecy, although the private key of the sender is disclosed, it does not affect the confidentiality of previous messages. By the public verification function, a judge directly verifies signature of original message without the sender's private key when dispute occurs. It enhances the justice of judge. The structure of this paper is organized as follows. Section 2 gives a background for HWSN networks. Section 3 introduces the proposed FS-ECSKM scheme. Section 4 analyses its security properties. Finally, section 5 concludes the paper.

## II. HWSN Routing Structure Background

### A. Cluster formation in HWSN

After sensor deployment, clusters are formed in an HWSN. For simplicity of discussion, assume that each H-sensor can communicate directly with its neighbor H-sensors (if not, then relay via L-sensors can be used). All H-sensors form a backbone in an HWSN. After cluster formation, an HWSN is divided into multiple clusters, where H-sensors serve as the cluster heads. An illustration of the cluster formation is shown in Fig. 1, where the small squares are L-sensors, large rectangular nodes are H-sensors, and the large square at the bottom left corner is the sink [1]. The routing structure in an HWSN is illustrated in fig. 1.

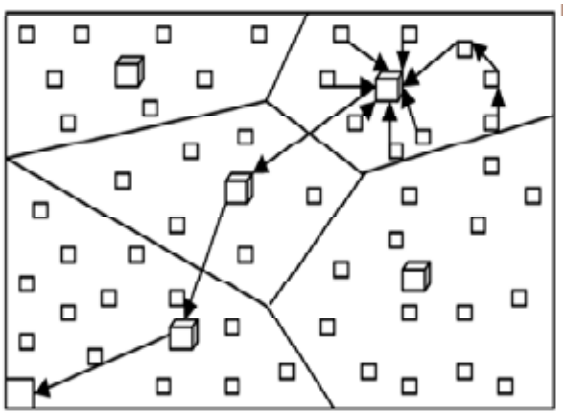

Fig. 1: Cluster formation in an HWSN [1]

### B. Routing in HWSN

In an HWSN, the sink, H-sensors and L-sensors form hierarchical network architecture. The HWSN consisting of two types of sensors [1]: a small number of high-end sensors (H-sensors) and a large number of low-end sensors (L-sensors). Both H-sensors and L-sensors are powered by batteries and have limited energy supply. Clusters are formed in the network and H-sensors serve as cluster heads. All H-sensors form a communication backbone in the network. Powerful H-sensors have sufficient energy supply, long transmission range, and high data rate. And thus provide many advantages for designing more efficient routing protocols. Routing in an HWSN consists of two phases:

- Intra-cluster routing- each L-sensor sends data to its cluster head via multi-hops of other L-sensors;
- Inter-cluster routing- a cluster head (an H-sensor) aggregates data from multiple L-sensors and then sends the data to the sink via the H-sensor backbone.

## III. Proposed Routing-Driven FS-ECSKM.

### A. Elliptic curve signcryption parameters

C: an elliptic curve [8] over GF ($p^m$), either with p $\geq 2^{150}$ and m = 1(public to all).
q: a large prime whose size is approximately of $|p^m|$ (public to all).
G: a point with order q, chosen randomly from the points on C (public to all).
Hash: a one-way hash function [9].
KH: a keyed one-way hash function [10].
(E; D): Encryption and Decryption algorithms of a private key cipher [11].

### B. Proposed protocol

Key setup for L-sensors can be achieved in either centralized or distributed way.

### 1. Centralized key establishment

We propose the following centralized EC Signcryption key management scheme. A server is used to generate pairs of ECC public and private keys, one pair for each L-sensor (and H sensor). The server selects an elliptic curve C over a large finite field GF and a point G on that curve. Each L-sensor (denoted as u) is pre-loaded with its private key (denoted as $V_L$ from $[1,\cdots,q-1]$), and with the public key of H-sensor ($P_H$). Each H-sensor is pre-loaded with a pair of common ECC public key ($P_H = V_H G$) and private key ($V_H$ from $[1,\cdots,q-1]$, and is pre-loaded also with public keys of all L-sensors (An H-sensor has large storage space).

The pre-loaded keys in H-sensors are protected by tamper-resistant hardware. Even if an adversary captures H-sensors, she could not obtain the key materials. Given the protection from tamper-resistant hardware, the same ECC public/private key pair can be used by all H-sensors, which reduces the storage overhead of the key management. H generates shared-keys for each L-sensor and its c-neighbors to make secure communication between them.

#### 1) Generate the Shared Key ($K_{sh_i}$)

This step is responsible for generating shared key between each L-sensor and its neighbour. H-sensor: Generate shared key $K_{sh_i}$ between each L-sensor and its c-neighbour, where $i = [1,\cdots,n]$, n: is the number of L sensors.

#### 2) Shared Key ($K_{sh_i}$) Signcryption

The H-sensor signcrypts the shared key ($K_{sh_i}$) using its private key and sends the ciphertext $(C_i, R_i, S_i)$ to the L-sensor as follows:

1. H-sensor chose a random $v_i \in [1,\cdots,q-1]$.
2. $k_{i,1} = hash\ (v_i G)$ (1)
3. $k_{i,2} = hash\ (v_i P_{L_i})$ (2)
4. $C_i = E_{k_{i,2}}\ (K_{sh_i})$ (3)
5. $r_i = hash\ (C_i, K_{i,1})$ (4)
6. $S_i = v_i/(r_i + V_H) \bmod q$ (5)
7. $R_i = r_i G$ (6)

The H-sensor sends the cipher text ($C_i, R_i, S_i$) to L-sensor; each L-sensor unsigncrypts the shared key ($K_{sh_i}$) as follows:

1. $k_{i,1} = hash\ (S_i(R_i + P_H)$ (7)
2. $r_i = hash\ (C_i, K_{i,1})$ (8)
3. $k_{i,2} = hash\ (v_{L_i} S_i(R_i + P_H)$ (9)
4. $K_{sh_i} = D_{k_{i,2}}(C_i)$ (10)
5. Accept $C_i$ only if $r_i G = R_i$ (11)

#### 3) The message between H-sensors signcryption

The H-sensor signcrypts the message m with its private key and send the signcrypted message to another H-sensor to make secure communication between them.

1. The H-sensor No.1 chose a random $v \in [1,\cdots,q-1]$.
2. $k_1 = hash\ (Gv)$ (12)
3. $k_2 = hash\ (Pv_H)$ (13)
4. $C = E_{k_2}(m)$ (14)
5. $r = hash\ (C, K_1)$ (15)
6. $S = v/(r + V_H) \bmod q$ (16)
7. $R =$ Gr (17)

The H-sensor No.1 sends the cipher text ($C$ , $R$ , $S$) to H-sensor No.2; the H-sensor No.2 unsigncrypt the message m as follows:

1. $k_1 = hash(S(R + P_H)$ (18)
2. $r = hash\ (C, K_1)$ (19)
3. $k_2 = hash(v_H S(R + P_H)$ (20)
4. $m = D_{k_2}(C\ )$ (21)
5. Accept $C$ only if Gr = R (22)

### 2. Distributed key establishment

The key setup can also be done in a distributed way. In the distributed key establishment, each L-sensor is pre-loaded with a pair of ECC keys- a private key and a public key. When an L-sensor (denoted as u) sends its location to its cluster head H. u signcrypts a message using its private key, and when H receives the message , it can unsigncrypt the message and then authenticate u's identify by using u's public key. After determining the routing tree structure in a cluster, the cluster head H signcrypts the tree structure (i.e., parent-child relationship) using its private key and disseminates it to each L-sensor. When each L-sensor receive the message containing the tree structure, it can unsigncrypt the

message by using H's public key and get the tree structure. If two L-sensors are parent and child in the routing tree, then they are c-neighbour of each other, and they will setup a shared key by themselves to start secure communication between them.

### 1) Location Message Signcryption

This step is responsible for signcrypting the location messages sending by each L-sensor to H-sensor. Then H unsigncrypt this messages and determines the tree structure.

Each L-sensor signcrypts location message with its private key $V_L$ as follows:

1. Each L-sensor chose a random $v_i \in [1,\cdots,q-1]$, Where $i=[1,\cdots,n]$, n: is the number of L-sensors.
2. $k_{i,1} = hash\ (v_i G)$ (23)
3. $k_{i,2} = hash\ (v_i P_H)$ (24)
4. $C_i = E_{k_{i,2}}(m)$ (25)
5. $r_i = hash\ (C_i, K_{i,1})$ (26)
6. $S_i = v_i / (r_i + V_{L_i}) \bmod q$ (27)
7. $R_i = r_i G$ (28)

Each L-sensor sends the cipher text $(C_i, R_i, S_i)$ to H-sensor; H-sensor unsigncrypts the message as follows:

1. $k_{i,1} = hash\ (S_i(R_i + P_{L_i}))$ (29)
2. $r_i = hash\ (C_i, K_{i,1})$ (30)
3. $k_{i,2} = hash\ (v_H S_i(R_i + P_{L_i}))$ (31)
4. $m = D_{k_{i,2}}(C_i)$ (32)
5. Accept $C_i$ only if $r_i G = R_i$ (33)

### 2) Broadcasting Message Signcryption

This step is responsible for signcrypting the broadcasting message from H-sensor to L-sensors authenticate the routing structure information. H sensor signcrypts the tree structure and disseminates it to each L-sensor.

1. H-sensor (CH) chose a random $v_i \in [1,\cdots,q-1]$, Where, $i=[1,\cdots,n]$, n: number of L sensors
2. $k_{i,1} = hash\ (v_i G)$ (34)
3. $k_{i,2} = hash\ (v_i P_{L_i})$ (35)
4. $C_i = E_{k_{i,2}}(m)$ (36)
5. $r_i = hash\ (C_i, K_{i,1})$ (37)
6. $S_i = v_i / (r_i + V_H) \bmod q$ (38)
7. $R_i = r_i G$ (39)

H-sensor sends the cipher text $(C_i, R_i, S_i)$ to each L-sensor; L-sensor unsigncrypts the message as follows:

1. $k_{i,1} = hash\ (S_i(R_i + P_H))$ (40)
2. $r_i = hash\ (C_i, K_{i,1})$ (41)
3. $k_{i,2} = hash\ (v_{L_i} S_i(R_i + P_H))$ (42)
4. $m = D_{k_{i,2}}(C_i)$ (43)
5. Accept $C_i$ only if $r_i G = R_i$ (44)

### 3) The message between L-sensors Signcryption

This step is responsible for signcrypting the messages between each L-sensor and its c-neighbour to start secure communication between them.

L-sensor u: send its public key $P_{Lv}$ to its c-neighbor.

c -neighbor v: send its public key $P_{Lv}$ to L-sensor u.

L-sensor u: Signcrypt the message m with its private key $V_L$.

1. The L-sensor u chose a random $v \in [1,\cdots,q-1]$.
2. $k_1 = hash(G)$ (45)
3. $k_2 = hash\ (v\ P_{L_v})$ (46)
4. $C = E_{k_2}(m)$ (47)
5. $r = hash\ (C, K_1)$ (48)
6. $S = v / (r + V_{L_u}) \bmod q$ (49)
7. $R = \mathrm{Gr}$ (50)

The L-sensor u sends the cipher text $(C, R, S)$ to c-neighbour v; the c -neighbour v unsigncrypt the message m as follows:

1. $k_1 = hash\ (S(R + P_{L_u}))$ (51)
2. $r = hash\ (C, K_1)$ (52)
3. $k_2 = hash\ (v_{L_v} S(R + P_{L_u}))$ (53)
4. $m = D_{k_2}(C)$ (54)
5. Accept $C$ only if Gr = R (55)

## IV. Performance Evaluation and Security Analysis

### A. Performance evaluation

In this section, the performance results for the proposed FS-ECSKM scheme has been discussed and compared with two schemes; ECC-based key management scheme given in [1] and ECS key management scheme given in [2] in terms of the storage requirement and energy consumption. It is shown that the proposed FS-ECSKM has the same performance compared with [2], but the proposed FS-ECSKM has an addition secure property, forward secrecy, public verification, and encrypted message authentication. The proposed FS-ECSKM scheme achieved better security with significant reductions on communication overhead, storage space, and energy consumption compared with [1]. Also, the proposed FS-ECSKM scheme has an addition secure property such as forward secrecy, public verification, and encrypted message authentication over the scheme given in [1].

### 1. Significant Storage Saving

Assume that the number of H-sensors and L-sensors in an HWSN is M and N, respectively. Typically we have M << N. In the Centralized EC signcryption key management scheme, each L-sensor is pre-loaded with its private key and the public key of H-sensors. Each H-sensor is pre-loaded with public keys of all L-sensors, plus a pair of private/public key for itself. Thus an H-sensor is pre-loaded with N+2 keys. Using EC compression point property in order to reduce the storage space, Hence, the total number of pre-loaded keys is:

$$\frac{M}{2} \times (N+2) + 2 \times N = (\frac{M}{2} + 2)N + M \quad (56)$$

In the distributed EC signcryption key management scheme, each L-sensor is pre-loaded with its public/private key pair and the public key of H-sensors. Each H-sensor is pre-loaded with public/private key pair and the public key of all L-sensors; also we use EC point compression. Thus the total number of pre-loaded keys is:

$$\frac{M}{2} \times (N+2) + 3 \times N = (\frac{M}{2} + 3)N + M \quad (57)$$

In the ECC-based key management scheme [1], the total number of pre-loaded keys in the Centralized ECC-based key management scheme is given by:

$$M \times (N+3) + 2 \times N = (M+2)N + 3M \quad (58)$$

The total number of pre-loaded keys in the Distributed ECC-based key management scheme is given by:

$$M \times (N+3) + 3 \times N = (M+3)N + 3M \quad (60)$$

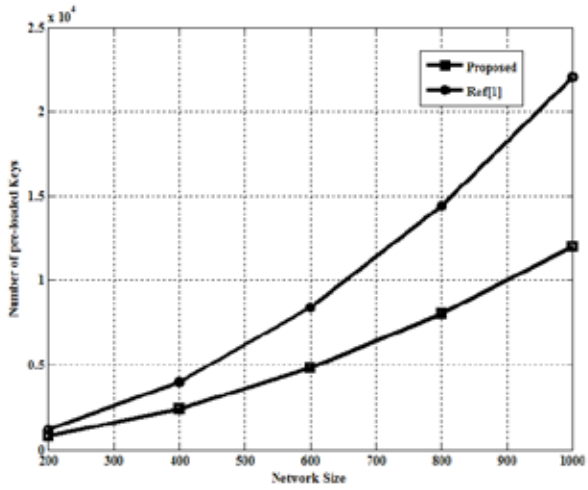


Fig. 2: Comparison with Centralized scheme given in [1]

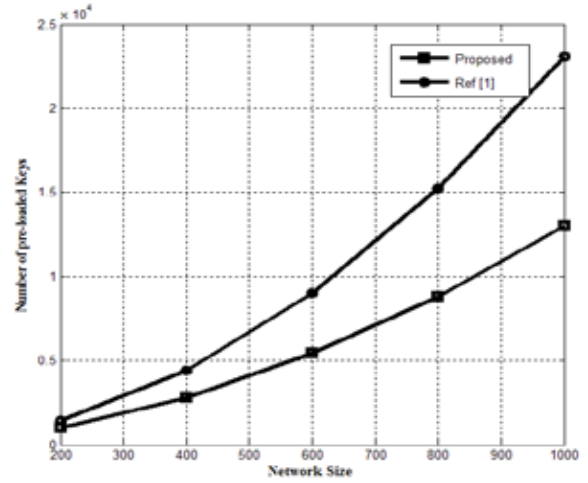


Fig. 3: Comparison with Distributed scheme given in [1]

In Fig. 2 and 3, the total number of pre-loaded keys for different sizes of sensor networks has been plotted for both centralized and distributed key establishment schemes. We can observe from Fig. 2 that the proposed scheme requires much less storage space for pre-loaded keys than that introduced in [1]. The results show that the proposed scheme provides better security than other schemes by offering the public verifiability and forward secrecy properties.

### B. Total Energy Consumption

Since the energy required transmitting 1 byte is 1.56μ Joule, so we can calculate the energy consumption for H-sensors. From the description of our protocol in the centralized scheme, we find that each H-sensor needs 56 bytes to send the routing structure information to each L-sensor, so the energy consumed can be computed as; Energy consumed per H-sensor = 56 * 1.56 = 87.36 μ Joule. It is the same value obtained when calculating the energy consumption in [1]. So, there is no saving in the energy consumption for this comparison but it is acceptable by achieving an addition secure property such as forward secrecy, public verification, and encrypted message authentication over the scheme given in [1].

### C. Security analysis

The security properties of the proposed scheme are described as follows [4]:

1. Unforgeability: It is computationally infeasible to forge a valid signcrypted text ( $C$ , $R$ , $S$ ) and claim that it is coming from the sender without having sender's private key.
2. Non-repudiation: If the sender denies that he sent the signcrypted text ( $C$ , $R$ , $S$ ), any third party can run the verification procedure below to check that the message came from him. Verification of $C$ , $R$ , $S$ by a judge or any third party:
   1. $k_1 = hash\ (S(R + P_{sender})$
   2. $r = hash\ (C, K_1)$
   3. Accept C only if Gr = R
3. Public verifiability: Verification requires knowing only sender's public key. All public keys are assumed to be available to all system users through a certification authority or a public directly. The receiver of the message does not need to engage in a zero-knowledge proof communication with a judge or to provide to prove.
4. Confidentiality: Confidentiality is achieved by encryption. To decrypt the cipher text, an adversary needs to have the receiver's private key.
5. Forward secrecy: An adversary that obtains the private key of the sender will not be able to decrypt past messages. Previously recorded values of ( $C$ , $R$ , $S$ ) that were obtained before the compromise cannot be decrypted because that adversary will need to calculate r to decrypt. Calculating r requires solving the ECDLP on R, which is computationally difficult.
6. Encrypted message authentication: The proposed scheme enables a third party to check the authenticity of the signcrypted text ( $C$ , $R$ , $S$ ) without having to reveal the plaintext m to the third party. This provides additional confidentiality in settling disputes by allowing any trusted/un-trusted judge to verify messages without revealing the sent message m to the judge.

## V. Conclusions

This paper presents an efficient Forward Secure Elliptic Curve Signcryption Key Management (FS-ECSKM) Scheme for Heterogeneous Wireless Sensor Networks (HWSN) has been proposed. An improved signcryption scheme that achieves the highly desired features in cluster sensor networks has been proposed. It utilizes elliptic curves for their high security and small key size. In addition, the new scheme achieves forward secrecy, public verifiability and encrypted message authentication. The scheme's forward secrecy property ensures that past messages remain confidential even if the sender's private key is compromised. Public verifiability enables any trusted/un-trusted judge when dispute occurs to verify the signature of original message without revealing any secret information. Also, the performance evaluation shows that this key management scheme can achieve significant reduction in storage space compared with other previous schemes.

## References

[1] Xiaojiang Du, Member, IEEE, Mohsen Guizani, Fellow, IEEE, Yang Xiao, Senior Member IEEE, and Hsiao-Hwa Chen, Senior Member, IEEE. "A Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks" vol. 8, March 2009.

[2] Esam A. A. A. Hagras, Hazem H. Aly, Doaa El-Saied "An Efficient Key Management Scheme based on Elliptic Curve Signcryption for Heterogeneous Wireless Sensor Networks". IJCST Vol. 1, Issue 2, December 2010.

[3] Y. Zheng, Digital Signcryption or how to achieve cost (signature and encryption)ii cost(signature) + cost(encryption)", In Advances in Cryptography – CRYPTO'97,LNCS 1294,

Springer-Verlag, pp. 165-179, 1997.
[4] Elsayed Mohamed and H. Elkamchouchi "Elliptic Curve Signcryption with Encrypted Message Authentication and Forward Secrecy" IJCSNS, Vol. 9 No.1, Jan. 2009.
[5] Sjouke Mauw, Ivo van Vessem, and Bert Bos "Forward Secure Communication in Wireless Sensor Networks" Springer- Verlag Berlin Heidelberg 2006.
[6] Yin Xin-Chun, Chun Jue-Wei, Wang C. Mei "A New Forward-Secure Signcryption Scheme" IEEE 2006.
[7] Ren-Junn Hwang, Chih-Hua Lai, Feng-Fu Su "An Efficient Signcryption Scheme with Forward Secrecy based on Elliptic Curve" Elsevier Inc. 2004.
[8] D. R. Hankerson, S. A. Vanstone, and A. J. Menezes, Guide to Elliptic Curve Cryptography, Springer, 2004.
[9] National Institute of Science and Technology, "Secure Hash Standard", USA, Federal Information Processing Standard (FIPS) 180-2, Aug. 2002.
[10] M. Bellare, "Keyed hash functions for message authentication", In Advances in Cryptology – CRYPTO'96 Vol. 1109, pp. 1-15.
[11] J. Daemen and R. Rijmen, "Rijndael: The Advanced Encryption Standard", Dr. Dobb's Journal, pp. 137- 139, Mar. 2001.

Esam A. A. HAGRAS received the B.S. degrees in Electrical Engineering from faculty of engineering, Alexandria Univ., Egypt, in 1994, M.S. degrees in Electrical Engineering from Mansoura Univ., Egypt, in 2001, respectively. During 2005-2007, he was on in Dept., of Electrical Engineering, faculty of engineering, Alexandria Univ. In Dec. 2007, he got the PhD degree in information security in communications. His research interests in the field of information and multimedia security, chaotic cryptography, Hardware implementation of encryption algorithms on FPGA ,data compression, digital image watermarking, communication and wireless sensor network security. He has published more than ten papers on security and communications.

Hazem H. Ali – Professor and Chairman of Communication Department, Arab Academy of Science & Technology and Maritime Transport, Egypt. He received the B.S. degrees in Electrical Engineering from faculty of engineering, Kuwait Univ., Kuwait, in 1987; M.S. and PhD degrees in Electrical Engineering from George Washington University, USA, in 1993 VLSI Systems and circuits. His research interests in the field of VLSI Systems and circuits, MEMs, NEMs, Hardware implementation of encryption algorithms on FPGA, communication and wireless sensor network security. He has published more than ten papers on security and communications.

Doaa El-Saied is currently a M.S. student in Department of Communication at Arab Academy of Science & Technology and Maritime Transport. She obtained her B.S. degree in 2000 from faculty of engineering in shoubra, Banha University, Egypt. Her research interests in security, communication systems, network security and wireless sensor networks.