Research Journal of Applied Sciences 13 (10): 571-581, 2018
ISSN: 1815-932X

An Intuitionistic Fuzzy Sets Implementation for Key Distribution in Hybrid Message Encryption Over WSNs

[1]Y.M. Wazery and [2]Mona A.S. Ali
[1]Faculty of Computers and Information, University of Minya, Minya, Egypt
[2]Faculty of Computers and Information, University of Benha, Benha, Egypt

Abstract: WSN is a way of handling dangerous and hostile environments safely. It replaces human existence with nodes and units that could sustain its existence under extreme circumstances. The significance of WSN arises from the importance of the data gathered through its nodes. Due to the fact of WSN that it is open air environment, security issues must be considered for example authentication of new units and the encryption of data transmitted between these units. This research provides a new model covering two important aspects in WSN. The first aspect is the creation of the key that will be used for the current session between a pair of nodes. In this step the research introduces the intuitionistic fuzzy sets to handle the WSN criteria simultaneously and efficiently in order to decide the exact key length required depending on the status of the network parameters. The second aspect is the distribution of the key between the units desiring communications. This phase starts by authenticating each entity to each other and to the cluster head, then one unit suggests a key and the other one confirms. It then starts communication using that key. This phase shows the hybrid cryptography applied in which the algorithm uses asymmetric encryption for authentication then uses symmetric encryption to secure the connection between the two units. Experimental results in this research could categorized also into two classes. The first class is key size model in which the proposed model compared to ordinary KNN and fuzzy model related to the determination of the key size. The proposed model shows an overall efficient way relating to decide the key size. The second class of experiments is to distribute the intermediate key efficiently at this point the proposed model shows resilience and efficiency compared to distributing the key directly through cluster head.

Key words: WSN, key, encryption, wireless, hybrid cryptography, communication, intuitionistic fuzzy sets, KNN

INTRODUCTION

The study introduces WSN (Wireless Sensor Network) with wide rang about its meaning and applications, it is not regarded as any ordinary network systems but it considers one of the most essential ways to introduce perfect and secure network service (Stallings and Tahiliani, 2014). To assure that we need to provide some circumstances and follow conditions which help users to access information in fields of interest of WSN easily without any obstructions or problems. In this study also encryption or encoding is recognized and implemented to provide a secure means of transmission and communication by knowing its origin, meaning and it's way of working in addition to its main purpose which protects data storage

In this research a new model for securing the WSN is proposed, the proposed model used to secure the creation and transmission of the secret key which is used for temporarily communication between a couple of entities. The creation of the temporarily depends on some parameters those are passed to an intuitionistic fuzzy model which decides the exact number of bits will be used under the current circumstances. After the number of bits is clearly decided, the model starts another phase in which that key is passed to a pair of units issued communication. The model starts with a communication request from a node to the cluster head. The CH authenticates each unit to the other then the session key is created and passed between the two units for a certain amount of time decided by the CH according to, the intuitionistic fuzzy model (Sri *et al*., 2017).

Literature review: WSN is the type of networks that is based on adhoc technology but provides more adequate and stable infrastructure. WSN provides, so, many applications both military and civilian environments. Security in WSN is a very hot issue for research, since,

Corresponding Author: Y.M. Wazery, Faculty of Computers and Information, University of Minya, Minya, Egypt

so, many attacks happen frequently, these attacks requires continuous development of defense systems to handle these attacks. The intuitionistic fuzzy sets provides a very elastic and strong methods for decision making within a very changing environment like WSN. The rest of this section provides insights on WSN, key distribution, cryptography and intuitionistic fuzzy sets. Those topics are the orbits for this research.

WSNs: A Wireless Sensor Network (WSN) means the wireless network which consists of spatially distributed over a range of autonomous devices by using sensors to provide the ability for controlling the environmental conditions. A WSN system incorporates a gateway that provides wireless connectivity back to the wired world and distributed nodes. The wireless protocol of choice works by depending on the application needs and requirements (Wazery and Abd-ELfattah, 2017).

Applications of WSN: Engineers create WSN applications for areas involving health care, utilities, surveillance and remote monitoring. In health care field, wireless devices create less invasive monitoring for patients (Begum and Srinivasan, 2016).

For services utilities such as the electricity power grid, streetlights and outdoor water municipals, wireless sensors give a lower-cost way for collecting system health data to decrease energy usage and better manage resources.

Remote controlling and control covers a wide range of applications including ways where wireless systems can sequel wired systems by decreasing costs of wiring structures and allowing many new types of measurement applications. Remote monitoring and surveillance applications include:

- Industrial large machine monitoring, saving human life from danger
- Structural monitoring for large buildings and bridges
- Environmental monitoring and assessment of air, soil and water (Hanafy *et al*., 2013)
- Process monitoring for watching over the steps involved in the automated processes without human intervention
- Important objects tracking

Wireless technology gives many advantages that help users to make wired and wireless systems and allow users to take advantage of the greatest technology for their applications.

Components of a WSN node: A WSN node has several technical ingredients involving the radio, battery microcontroller, sensor interface and analog circuit. When

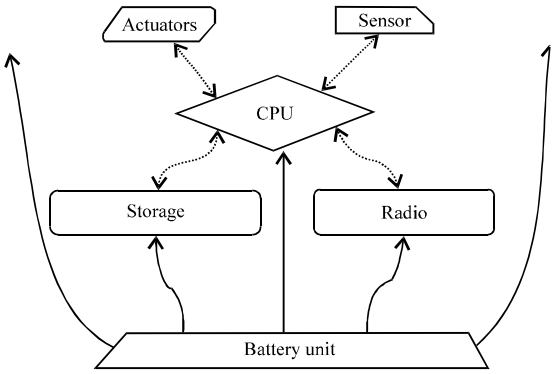

Fig. 1: General form of WSN

microcontroller, sensor interface and analog circuit. When WSN technology is used, trade-offs among those composts must be kept in mind. In systems those are mainly battery-powered, the use of more frequent radio besides higher radio data rates implies more power consumption. Usually 2 and 3 years battery life is required, so, most of the WSN systems today are built on ZigBee because of the low-power consumed in Zigbee, due to battery life and power management technology are evolving and due to the availability of IEEE 802.11 bandwidth, Wi-Fi will be an interesting technology (Al-Shehri, 2017).

The other technology requirement in WSN architecture is the battery itself. In addition to long life required, the size and weight of batteries must be considered as well as internationally, existing standards for the shipping of batteries and the availability of battery. The low cost and wide availability of carbon zinc and alkaline batteries make them a common choice (Fig. 1).

To enlarge the battery life, a WSN node continuously wakes up and transmits data by powering on the radio and then powering it back off to keep energy. WSN radio technology must transmit a signal efficiently and allow the system to go back to sleep with lower power use. This means the processor included must also be able to power up, wake and return to sleep mode in an efficient way. WSNs microprocessor direction involve reducing consumption of power while reserving or increasing processor speed. Much like any radio choice, the processing speed and power consumption trade-off is a key issue and concern when selecting WSNs processors (Garg, 2017). This makes processors of the family ×86 architecture a very hard option for any battery-powered units.

Intuitionistic fuzzy sets: Intuitionistic Fuzzy Set (IFS), introduced by Atanassov is considered a powerful tool to handle deal with vagueness. A prominent obvious

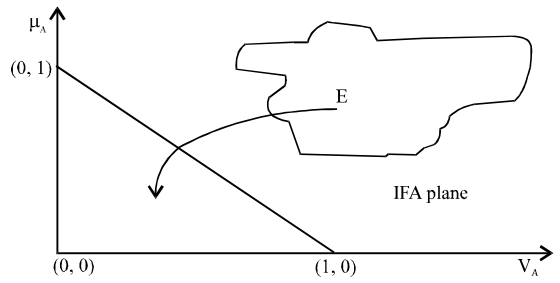

Fig. 2: Space allocation of IFS

characteristic of IFS is that it assigns to each element first a membership degree and secondly a non-membership degree and thus, the IFS constitutes an advancement and extension of Zadeh's fuzzy set which assigns only a membership degree to each element (Ford *et al*., 2017). Many researchers have paid attention for the applications of the IFS theory. Those applications and theories has been successfully used and applied in different fields such as; Logic programming (Hanafy *et al*., 2012), medical diagnosis, decision making problems etc. Recently various applications of IFS clustering and classification of artificial intelligence have appeared for example (IFNN) Intuitionistic Fuzzy Neural Networks, (IFES) Intuitionistic Fuzzy Expert Systems, (IFML) Intuitionistic Fuzzy Machine Learning (Hanafy *et al*., 2012). (IFDM) Intuitionistic Fuzzy Decision Making (IFSR) Intuitionistic Fuzzy Semantic Representations, etc. intuitionistic fuzzy sets (Fig. 2).

Let a set E be fixed. An IFS A* in E is an object having the form: $A^*=\{\langle x,\mu_A(x),V_A(x)\rangle \mid x\in E\}$ where the functions $\mu_A(x)$: $E \rightarrow [0, 1]$ and $v_A(x)$: $E \rightarrow [0, 1]$ define the degree of membership and the degree of non-membership of the element $x\in E$ to the set A which is a subset of E (for simplicity below, we shall write A instead of A*), respectively and for every $x\in E$: $0\leq \mu_A(x)+v_A(x)\leq 1$.

Key distribution: Key distribution might be defined as the process of distributing (cryptographic) keys to different parties. Usually this distribution includes techniques regarded "Out-of-band", i.e., techniques that don't use the channel again of later connections to transmit keys. Alternative method for key distribution can be achieved through the relying of the distributing new keys onto the safe distribution of old keys that's what a KDC is doing (Abdelmged *et al*., 2016).

The standard meaning for distributing keys propose administration over the entire lifetime of the key. Key management and distribution is a piece of key administration, however, it additionally includes key creation, key escrow (for reinforcement purposes), key erasure, key repudiation, key utilization and key trust in administration.

Cryptography is likewise used to help the procedures for validating entities between sets of nodes. Authentication rules and protocols are about dissemination and administration of secret keys (Ejegwa *et al*., 2014).

Key management and distribution in an appropriated environment is a usage of dispersed verification protocols. Based on this thought many key dissemination and verification conventions have been proposed.

Generally, all protocols and mechanisms expect that some secret data is held at first by every management unit. Authentication and verification is accomplished by one central node exhibiting the other that manages that key. All frameworks accept that strategy condition is exceptionally unstable and is open for assault.

So, any message arrived from a central unit must have its authentication, integrity and freshness confirmed. To accomplish these objectives, most frameworks need to depend on a confirmation server and this server ought to have the accompanying highlights (Bhattacharya, 2016).

Ability: An authentication server conveys great quality session keys and disperses them to the asking for principals safely.

Trustability: Authentication server keeps up a table containing a name and a private key for every unit. The secret key is utilized just to confirm unit's actions to the verification server and to transmit messages safely between customer forms and the confirmation server (Hanafy *et al*., 2012).

Key distribution and validation protocols are isolated into two classifications to assure the confirmation of a message. To begin with class utilizes nonce and test/reaction handshake to check freshness, illustration is Needham-Schroeder protocol. Second classification utilizes timestamps and expect that all machines in appropriated framework are clock-synchronized case is Kerberos protocol (Houssein and Ismaeel, 2015).

Encryption: Encryption is the transformations of electronic data from a form to another, aka cipher text which difficult to be understood or decrypt by anyone except authorized units. The main reason and goal of encryption is to preserve the secrecy and confidentiality of digitally stored or transmitted data files through the internet or other computer networks used. Modern ciphering algorithms play a truly crucial part in the security and assurance of electronic communication systems as they can preserve both confidentiality and the following vital key security elements (Praveena and Smys, 2016).

Algorithm 1; Key size generation by using intuitionistic fuzzy model:

Input: Node Count (NC), Node Log (NL), Trusted Neighbors Count (TNC), Frequency of key Changes (FKC) and Length of Temporarily Key (LTK)
Output: length of the intermediate encryption key (in bits) (IEK)
For each input variable

|Calculate μA(X) ~ ∨A(X)
|Apply Intuitionistic fuzzy system conductive IF-Then rules

End

Algorithm 2; Key distribution:

Input: intermediate encryption key (in bits) (IEK)
Output: Key distributed
// Stage 1
1. X sends a message to CH $C_{to}(CH)=E_{KU(CH)}$ ($E_{KR(X)}$(id , Y))
2. CH receives message and analyze $M_{frm}(X) = D_{KR(CH)}$ ($E_{KU(X)}$(id, Y))¬(id, Y)
3. If decryption error or id is obsolete:
   |Increase NL(X)//consider attack
   |Discard the message
   Else search entire DB for KU(Y)
4. If KU(Y) does not exist
   |Send an error message to X
   Else Send a message to X contains Y's KU $C_{to}(X)$ = ($E_{KU(X)}$ ($E_{KR(CH)}$(id, KU(Y)))
5. CH receives Message and analyze $M_{frm}(CH) = D_{KR(CH)}$ ($E_{KU(X)}$(id , KU(Y)))¬id, KU(Y)
6. If decryption error or id is obsolete
   |Discard the massege
   Else Store Y's public key and pass it to stage 2
// Stage 2
7. X sends a message to Y $M_{to}(Y)$ = (id , KU(X)))
8. Y receives Message and analyze id , KU(X)
9. If id is obsolete:
   |Discard the massege
   Else Y sends a query to CH $C_{to}(CH)=E_{KU(CH)}$ ($E_{KR(Y)}$(id , KU(X)))
10. If decryption error or id is obsolete
   |Discard the message
   |Increase NL(Y)//consider attack
   Else search entire DB for KU(X)
11. If KU(X) does not exist
   |Send an error message to Y
   Else send a message to Y contains X's KU $C_{to}(Y)$ = ($E_{KU(Y)}$ ($E_{KR(CH)}$(id, KU(X)))
12. Y confirms X's public key from CH's message $M_{frm}(CH) = D_{KR(CH)}$ ($E_{KU(Y)}$(id , KU(X)))
13. If decryption error or id is obsolete:
   Discard the message
// Stage 3
14. Y sends a confirmation to X $C_{to}(X)$ = ($E_{KU(X)}$ ($E_{KR(Y)}$(id , KU(X)))
15. X decrypt Y's message id , KU(X)
16. If id is obsolete:
   Discard the message
   Else
   X sends the intermediate encryption key to Y $C_{to}(Y)$ = ($E_{KU(Y)}$ ($E_{KR(X)}$(id , IEK))
17. Y extracts the IKE of the message from X Mfrm(X) = DKR(Y) (EKU(X)(id , IEK))¬(id, IEK)
18. If decryption error or id is obsolete:
   Discard the message
   Else store IEK
19. Y sends a hello message encrypted with IEK $C_{to}(X) = E_{IEK}$ (id,"HEllo")
20. X decrypts the message obtaining the id and "HEllo" confirming the IEK

Authenticity: The source of a message could be assured and confirmed. Integrity: a proof of the case that message contents have been modified or not since its transmission. Non-repudiation: the originating unit of a message cannot claim that the message does not belong to him (Wazery and Abd-ELfattah, 2017).

Symmetric ciphers (single key encryption) are the type of encryption in which all entities share one secret key for both ciphering and deciphering a file. AES is considered to be one of the most widely used single-key encryption (Hanafy *et al*., 2012). Symmetric encryption provides a faster processing than asymmetric-key encryption but with a drawback that the sender must somehow transforms the secret key used to encrypt the data to the other unit (s) before start using that key. This is a basic condition to securely manage and distribute huge amounts of symmetric keys implies that most cryptography models use a symmetric encryption algorithm to cipher data efficiently on the other hand they use asymmetric encryption algorithm for the purposes of secret key transmission (Elsawy and Osamy, 2016).

Asymmetric (double-key) cryptography or public-key cryptography this type of ciphering uses a couple of mathematically related but different keys. The first key is public (available to some or all other units) and the other key is to be private (secret). One of a commonly used asymmetric encryption is RSA algorithm, basically because both keys (public and private) can be used to encipher a transmission in the same time only the other key from the one used to encipher a message can be used to decipher it. This condition allows a trust way of assuring not only integrity but also the confidentiality, non-reputability and authenticity of an electronic connection (Ejegwa *et al*., 2014).

MATERIALS AND METHODS

The proposed paradigm composes of two stages the first is to determine the size of the intermediate encryption key, the second stage is to distribute the keys. Both stages will be clearly identified then finally a pseudocode of the entire process shall be driven.

Intuitionistic Fuzzy Sets (IFS) Model: For the purpose of providing the accurate size of the intermediate encryption key, the algorithm must keep track of the rapidly changing parameters in the WSN. Hence, the expected level of security depends mainly on the difficulty of breaking the secret key between each pair of communicating devices. The crucial step in the proposed model is the design of the IFS function that produces the actual size of the session key (intermediate encryption key) by processing the parameters given. This process handles five variables (Nodes Count (NC), Node Log (NL), Trusted Neighbors

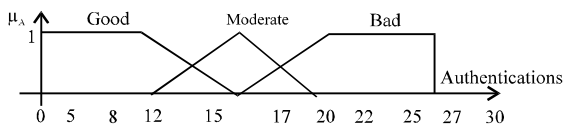

Fig. 3: NL IFR (μ_A)

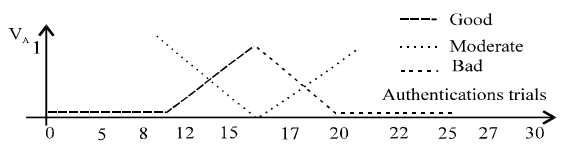

Fig. 4: NL IFR (V_A)

Count (TNC), Frequency of Key Changes (FKC) and Length of Temporarily Key (LTK)) each of those variables will have a degree of membership μA(x) and a degree of non-membership vA(x) as illustrated below:

Nodes Count (NC): A counter intuitionistic fuzzy variable that holds the number of nodes currently registered to the WSN taking two fuzzy values:

- Small with μ_A(sml) and v_A(sml)
- Many with μ_A(ma) and v_A(ma)

Node Log (NL): An intuitionistic fuzzy variable that monitors the history of the node's authentication attempts, this variable takes three values:

- Good: the node had been registered many times and causes no susceptibility with μ_A(Go) and v_A(Go)
- Moderate: in states that the node had been registered many times but causes small number of susceptibilities with μ_A(Mod) and v_A(Mod)
- Bad: declares that the node is a potential risk either by its self or through a BOTnet attack with μ_A(Bad) and v_A(Bad)

The node log variable will be expressed graphically as follows in Fig. 3 and 4:

Trusted Neighbors Count (TNC): Counter intuitionistic fuzzy variable that keeps track of the number of neighbors with a certain threshold of distance from the node A. that variable takes three values:

- Little: indicates that the number of neighbors is small hence the amount of attacks is relatively small, this variable is associated with two states μ_A(Lit) and v_A(Lit)

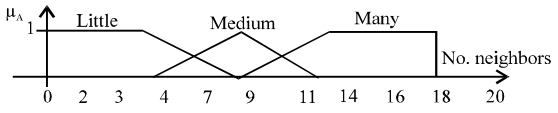

Fig. 5: TNC IFR (μ_A)

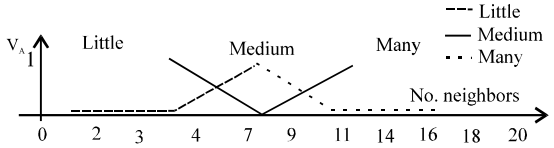

Fig. 6: TNC IFR (V_A)

- Medium: indicates that the number of neighbors is small hence the amount of attacks is relatively medium, this variable is associated with two states μ_A(Med) and v_A(Med)
- Many: indicates that the number of neighbors is small, hence, the amount of attacks is relatively large, this variable is associated with two states μ_A(Man) and v_A(Man) (Fig. 5 and 6).

Frequency of Key Changes (FKC): The frequency of changing the session key which is an intuitionistic fuzzy variable, the more frequent changing key is for sure safer and provides more resilience to the security overall but still creates more processing and more resources consumption. This variable handles two values:

- S: ordinary traffic flow and small number of changes to the session key, this variable is associated with two states μ_A(S) and v_A(S)
- F: high traffic and fast changing the session key, this variable is associated with two states μ_A(F) and v_A(F)

Length of Temporarily Key (LTK): The length of the temporarily session key which is also an intuitionistic fuzzy variable for outputting the key length. This variable handles three values:

- S: the fewest number of bits for a session key usually 16:64 bits depends on the inputs
- M: moderate number of bits for a session key usually 64:184 bits depends on the inputs
- L: large number of bits for a session key usually 184:512 bits depends on the inputs

The reason to handle and process those variable is to obtain the desired security by find the exact value of another variable Session Key Scale (SKS) variable with the values ranging (very low, low, normal, high, very high) the table below illustrates the intuitionistic rules applied in each case.

Table 1: IFS inputs and outputs

Intuitionistic fuzzy inputs					Output	Output
NC	NL	TNC	FKC	LTK	SKS	Number of bits
μ_A(sml)~v_A(sml)	μ_A(Go)~v_A(Go)	μ_A(Lit)~v_A(Lit)	μ_A(S)~v_A(S)	S	Very low	12-16
μ_A(ma)~v_A(ma)	μ_A(Mod)~v_A(Mod)	μ_A(Med)~$_A$(Med)	μ_A(F)~v_A(F)	M	Low	24-32
μ_A(sml)~v_A(sml)	μ_A(Bad)~v_A(Bad)	μ_A(Man)~v_A(Man)	μ_A(S)~v_A(S)	L	Normal	48-64
μ_A(sml)~v_A(sml)	μ_A(Mod)~v_A(Mod)	μ_A(Man)~v_A(Man)	μ_A(F)~v_A(F)	M	Normal	48-64
μ_A(ma)~v_A(ma)	μ_A(Go)~v_A(Go)	μ_A(Lit)~v_A(Lit)	μ_A(S)~v_A(S)	S	Low	24- 2
μ_A(ma)~v_A(ma)	μ_A(Go)~v_A(Go)	μ_A(Med)~ví$_A$(Med)	μ_A(F)~v_A(F)	S	High	128-160
μ_A(sml)~v_A(sml)	μ_A(Mod)~v_A(Mod)	μ_A(Lit)~v_A(Lit)	μ_A(S)~v_A(S)	L	Normal	48-64
μ_A(sml)~v_A(sml)	μ_A(Go)~v_A(Go)	μ_A(Man)~v_A(Man)	μ_A(F)~v_A(F)	L	High	128-160
μ_A(ma)~v_A(ma)	μ_A(Bad)~v_A(Bad)	μ_A(Med)~v_A(Med)	μ_A(S)~v_A(S)	M	Very high	256-300
μ_A(sml)~v_A(sml)	μ_A(Bad)~v_A(Bad)	μ_A(Man)~v_A(Man)	μ_A(F)~v_A(F)	M	Normal	48-64
μ_A(ma)~v_A(ma)	μ_A(Go)~v_A(Go)	μ_A(Lit)~v_A(Lit)	μ_A(S)~v_A(S)	S	Very high	256-300
μ_A(ma)~v_A(ma)	μ_A(Mod)~v_A(Mod)	μ_A(Med)~v_A(Med)	μ_A(F)~v_A(F)	L	High	128-160
μ_A(sml)~v_A(sml)	μ_A(Bad)~v_A(Bad)	μ_A(Lit)~v_A(Lit)	μ_A(S)~v_A(S)	M	Low	24-32
μ_A(ma)~v_A(ma)	μ_A(Mod)~v_A(Mod)	μ_A(Man)~v_A(Man)	μ_A(F)~v_A(F)	L	High	128-160
μ_A(ma)~v_A(ma)	μ_A(Go)~v_A(Go)	μ_A(Med)~v_A(Med)	μ_A(S)~v_A(S)	S	Normal	48-64

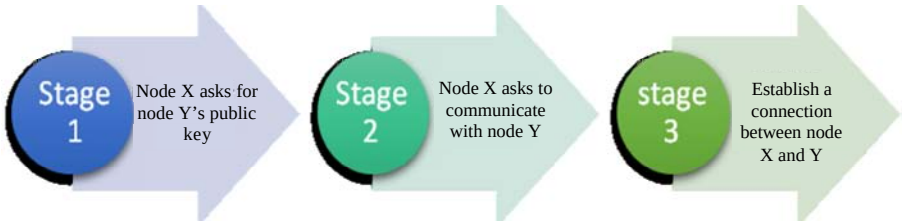

Fig. 7: Key distribution model

Table 1 provides the basis for the fuzzificaction process implementing the IF-then, rules and all of the other steps in the process. The value of the variable SKS determines the fuzzy value for the session key length. The defuzzification process shown in the Table 1 which provides the actual length of the session key.

Key distribution model: After deciding the suitable length of the intermediate encryption key, this length must sent in a secret way to both nodes that asking for communication. In order to do, so, a way of asymmetric encryption must be used. For the purpose of asymmetric encryption RSA was preferred for the implementation to obtain the public key and private key for each node within the WSN. These keys are distributed within the shake hand protocol when the node is first registering to the WSN, the hand shake protocol is to be done with the Cluster Head (CH) which is one node responsible for keeping track of each node within the WSN in its range if this CH tries to leave the WSN or goes down for any reason it transmits all of its control information and data bases to a node with the highest node log in the WSN. Now another problem arises which is the way of distributing the intermediate encryption key. To distribute the session keys safely means of hierarchal communication must be used. One of the most famous and practical solution for the hierarchal communication is the PKI. A customized version of the PKI is used in this algorithm taking three stages as shown in Fig. 7.

Stage 1; Node X tries to gain access to the public key of node Y, this can be done over four steps

Step 1 (X requests Y's public key from CH): Node X sends a request message to the CH, requiring the public key of node Y this message is in the form C_{to} which means cipher to (CH):

$$C_{to}(CH)=E_{KU(CH)}\left(E_{KR(X)}(id,Y)\right) \quad (1)$$

This message contains the id of the message to prevent replay attacks and the name of the node Y. The message is encrypted two times first with the private key of node X (to assures that it come from node X, since, it cannot be decrypted by any key except by X's public key which is available to the CH) then the message is encrypted with CH's public key which is available to every node in the WSN. The reason behind the last encryption to assure that only CH can open the message by its private key.

Step 2 (CH decrypt X's message): CH receives the message from X and analyze it in the form of reversing the order of the message $M_{frm}(X)$ which message from X. CH first decrypts by its own private key then with Y's public key:

$$M_{frm}(X)=D_{KR(CH)}\left(E_{KU(X)}(id,Y)\right)\rightarrow(id,Y) \quad (2)$$

Step 3 (CH sends Y's public key to X): CH encrypts a message in the form:

$$C_{to}(X)=(E_{KU(X)}(E_{KR(CH)}(id, KU(Y)))) \quad (3)$$

Double encryption in this step serves as encryption of the message and authentication of (CH).

Step 4 (X retrieve Y's public key from CH's message): X decrypts the message as the following equation:

$$M_{frm}(CH)=D_{KR(CH)}(E_{KU(X)}(id, KU(y)))\rightarrow id, KU(Y) \quad (4)$$

Stage 2: Node X asks node Y for communication

Step 1 (X sends a communication request to Y): X sends a non-encrypted message to Y including X's public key and an id to declare the timing of the message:

$$M_{to}(Y)=(id, KU(X)) \quad (5)$$

Step 2 (Y sends a query to CH): Node X sends a request message to the CH, requiring the public key of node Y this message is in the form C_{to} which means cipher to (CH):

$$C_{to}(CH)=E_{KU(CH)}(E_{KR(Y)}(id, KU(X))) \quad (6)$$

The message includes the id of the message to prevent replay attacks and the public key of the node X. This message is encrypted two times first with node Y's private key to authenticate Y to CH. Then the message is encrypted with CH's public key that prevents any other node from reading the content of the message.

Step 3 (CH decrypt Y's message): CH receives the message sent from Y and analyzes in the form of reversing the order of the message $M_{frm}(Y)$ which message from Y:

$$M_{frm}(Y)=D_{KR(X)}(E_{KU(Y)}(id,KU(X)))\rightarrow(id,KU(X)) \quad (7)$$

Step 4 (CH sends X's public key to Y): CH encrypts a message containing the public key of X as a confirmation to Y in the form:

$$C_{to}(Y)=(E_{KU(Y)}E_{KR(CH)}(id, KU(Y))) \quad (8)$$

Step 5 (Y confirms X's public key from CH's message): Y decrypts the message as the following Eq. 9:

$$M_{frm}(CH)=D_{KR(CH)}(E_{KU(Y)}(id, KU(X)))\rightarrow id, KU(X) \quad (9)$$

Up to this step X and Y are both confirmed to each other and to CH that leads the flow to stage 3 in which they are communicating through a common shared key with the size given from the intuitionistic fuzzy model for key distribution that is mentioned earlier.

Stage 3: Establish a connection between X and Y

Step 1 (Y sends a confirmation to X): In order to inform X that Y got the confirmation about X from CH, Y sends a message in the form:

$$C_{to}(X)=(E_{KU(X)}(E_{KR(Y)}id, KU(X))) \quad (10)$$

This message contains the public key of X to confirm the acknowledgment about X this message will be encrypted two times first encrypted with the private key of the Y node to authenticate Y-X, then the message is encrypted with X's public key that prevents any other node from reading the content of the message.

Step 2 (X decrypt Y's message): X receives the message sent from Y and analyze in the form of reversing the order of the message = $M_{frm}(Y)$ which message from Y:

$$M_{frm}(X)=D_{KR(CH)}(E_{KU(Y)}(id, KU(X)))\rightarrow(id, KU(X)) \quad (11)$$

Now both X and Y are confirmed and authenticated to each other, so, they can communicate with a common shared key that will be used for any symmetric encryption technique.

Step 3 (X sends the intermediate encryption key to Y): In order to start a private session between X and Y X sends an intermediate session key (IKE) to Y:

$$C_{to}(Y)=(E_{KU(Y)}(E_{KR(X)}(id, IEK)) \quad (12)$$

This message contains the IEK this message will be encrypted two times first encrypted with the private key of the X node to authenticate X-Y, then the message is encrypted is encrypted with Y's public key that prevents any other node from reading the content of the message.

Step 4 (Y extracts the IKE of the message from X): Y receives the message sent from X and analyze in the form of reversing the order of the message $M_{frm}(Y)$ which message from Y:

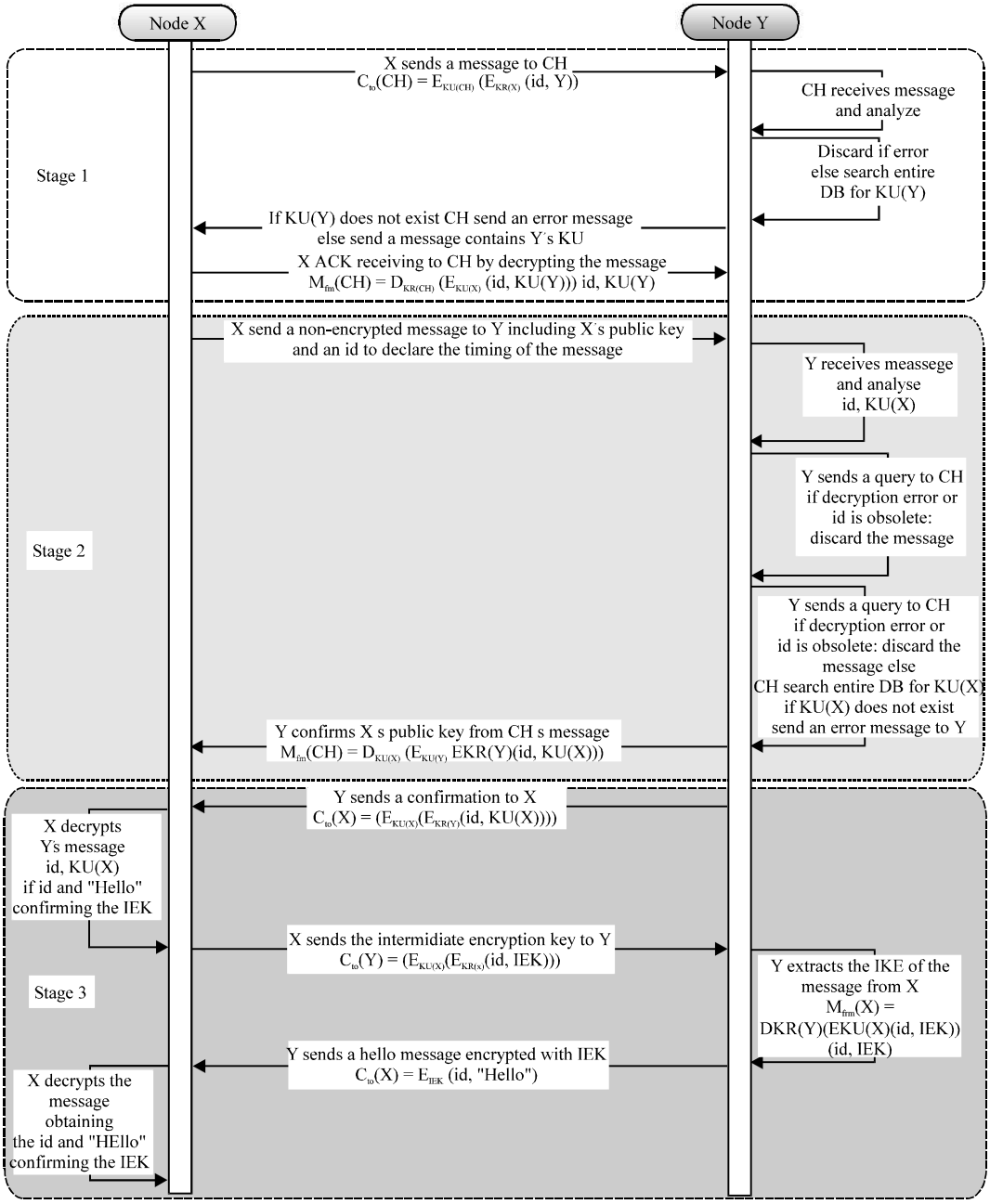

Fig. 8: Sequence diagram for communication

$$M_{frm}(X)=D_{KR(Y)}\left(E_{KU(X)}(id, IEK)\right)\rightarrow(id, IEK) \quad (13)$$

Step 5 (Y sends a hello message encrypted with the intermediate encryption key to X): The final step before both nodes can start their communication using IEK is that Y sends a message encrypted using the symmetric encryption algorithm using the IEK:

$$C_{to}(X)=E_{IEK}(id,"HEllo") \quad (14)$$

By the time this message arrives to X. It is confirmed that Y got the IEK, so, they can start their session using that IEK (Fig. 8).

RESULTS AND DISCUSSION

During the implementation of this research a set of experiments performed and a set of experimental results recorded. The first type of experiments was to discriminate among different tools and methods of deciding the size of

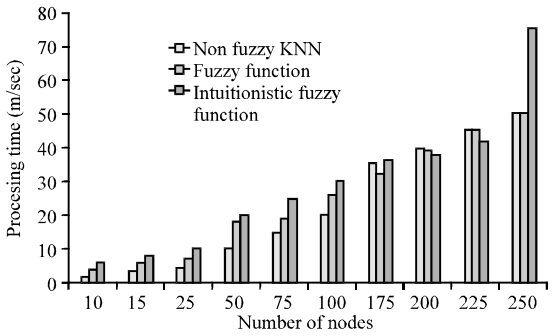

Fig. 9: ASL of intuitionistic fuzzy vs. non-intuitionistic fuzzy classification

the intermediate session key as the other set of experiments was in the key distribution methods was clarified. This section will compare the results of applying the proposed model to some algorithms and models used to provide the same functionality. To compare different type of algorithms two criteria are used in each case.

Processing time: The time required to process the nodes under consideration by the algorithm, in other words the time required to evaluate the mathematical and logical operations required by the algorithm.

Gained security level: The level of security provided by each algorithm this criteria will be measured through monitoring two indicators:

- False rejection: number of non-attacker nodes which the model rejected or eliminated
- False acceptance: number of attacker nodes which the algorithm did not discovered

Intuitionistic fuzzy model (key size): The key size determination function is the first step in the proposed model. The results of the intuitionistic implementation were compared to two will know algorithms in the field, KNN and fuzzy implementation. Results were monitored and recorded in each case regarding to the two applied criteria.

Processing time: The it takes the model to generate the key is referred to as the processing time, it differs from environment to another, for example, some applications are delay tolerant like mail transmission and FTP where some other environments does not provide any tolerance for delay such as real time environment. The concern of this research is on the first type delay tolerant applications. In the same time the processing time must be bounded and reasonably accepted in order to keep the processing capabilities intact and healthy. These results are in milliseconds and shown in Fig. 9.

Table 2: False acceptance of intuitionistic fuzzy vs. fuzzy function vs. KNN

	False acceptance		
No. of nodes	Non fuzzy KNN	Fuzzy function	Intuitionistic fuzzy function
10	2	2	1
15	5	5	3
25	8	9	4
50	12	11	7
75	18	17	8
100	25	22	10
175	45	39	17
200	55	48	22
225	60	52	25
250	70	67	90

Table 3: False reject of intuitionistic fuzzy vs. fuzzy function vs. KNN

	False reject		
No. of nodes	Non fuzzy KNN	Fuzzy function	IntuitionisticFuzzy function
10	1	2	1
15	2	3	2
25	3	4	2
50	6	7	3
75	9	8	4
100	10	9	4
175	15	11	6
200	20	14	7
225	23	16	9
250	18	19	9

Figure 9 shows the processing time for the intuitionistic fuzzy function is larger for the small number of units which is logically due to the number of parameters handled and the computation time required but as the system grows up and the number of units increases the proposed model outperforms the two other algorithms with respect to the processing time.

Gained security level: The gained security levels for three types are measured in terms of false acceptance and false rejections and shown in Table 2 and 3.

Table 2 and 3 show how intuitionistic fuzzy decision making for the size of the key outperforms the other two models with a very significant ratio in both false acceptance and false rejections these superiorities condones the time delay obtained for the small number of units.

Key distribution model: The most important step in the proposed algorithm which is the decision of the key size is done with its experiments and proven that the intuitionistic fuzzy model outperforms the other two alternatives. Now, it is the time to examine the next stage in the model which is the distribution of the key. One way is to provide each of the units with IEK directly through the CH which will be referred as DCH and the other alternative is to use the infrastructure provided by the algorithm shown in the proposed model which will be referred as IPM. In this study both paradigms will be compared in terms of the two common criteria.

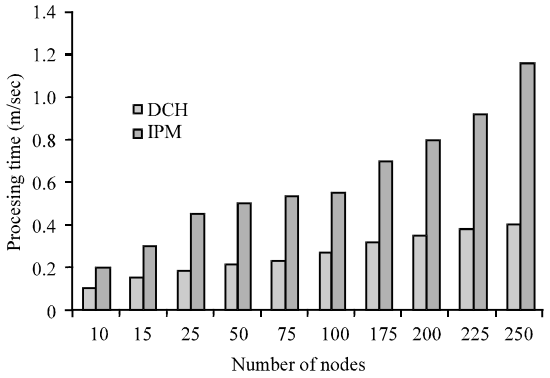

Fig. 10: Security of DCH vs. IPM

Table 4: Flse acceptance of DCH vs. IPM

	False acceptance	
No. of nodes	DCH	IPM
10	1	1
15	3	3
25	8	4
50	15	6
75	21	11
100	30	14
175	48	18
200	57	19
225	62	21
250	65	23

Table 5: False acceptance of DCH vs. IPM

	False reject	
No. of nodes	DCH	IPM
10	1	1
15	2	2
25	4	3
50	8	4
75	10	6
100	12	7
175	18	7
200	22	8
225	25	8
250	30	9

Processing time: The processing time for both mechanisms is measured in milliseconds and shown in Fig. 10. Figure 10 shows one weakness of the proposed model that it takes more time to distribute the key in fact this amount of time as it is really obvious it is still negectable regarding to the security provided by the proposed model.

Gained security level: One other important criterion that must be considered while developing any solution for a security model that is the gained security of the implementation, since, the algorithm is capable of deciding whether to accept or reject the node and alter its log variable, then, the gained security has to be in consideration. Next Table 4 and 5 shows the number of false acceptance and reject shown by implementing key distribution scenarios.

Table 4 and 5 shows that the IPM provides a far more enhanced security levels than DCH in both cases, either the decision to accept a node or to reject it. Specifically, when it comes to rejection IPM provides semi stable performance as the number of nodes continues to grow other than the decaying performance of DCH which provides more false reject and acceptance than IPM.

CONCLUSION

The importance of the WSN comes from the fact that it could replace human existence in dangerous and hostile environment. WSN provides a scalable, easy to implement and very resilient infrastructure. This research provides a novel model for securing the WSN, via. making sure that the intermediate encryption key is used for short periods and distributed safely. The model illustrated could be viewed as a two phase's paradigm, the first phase decides the number of bits required for the IEK according to the current network conditions, this number of bits is to be passed to the second phase which is the key distribution model. In the later phase each unit tries to make sure that it will communicate with the right unit, this done through securely communicate to a trusted third unit which is called CH. Experimental results performed in this research could be also categorized into two categories the first handles the key size determination in which the research shows that the use of intuitionistic fuzzy model will increase the security levels significantly where as it provides a little bit of increment in processing time but this amount of time is relatively small compared to the advancement in security levels. The other category of the experimental results handles the IEK distribution model in this step IPM shows advancement in the security levels that overcomes the delay it provides regarding to the processing time.

RECOMMENDATION

For future research, we are looking forward to apply some colony classification such as bee or whale to decide the key size. Additionally we will implement elliptic curve strategy for key distribution.

REFERENCES

Abdelmged, A.A., A.H.S. Saad and N. Hussien, 2016. A combined approach of steganography and cryptography technique based on parity checker and huffman encoding. Intl. J. Comput. Appl., 148: 26-32.

Al-Shehri, W., 2017. A survey on security in wireless sensor networks. Intl. J. Netw. Secur. Appl., 9: 25-32.

Begum, S.S. and R. Srinivasan, 2016. A study on properties of intuitionistic fuzzy sets of third type. Intl. J. Math. Appl., 4: 59-64.

Bhattacharya, J., 2016. A few more on intuitionistic fuzzy set. J. Fuzzy Set Valued Anal., 2016: 214-222.

Ejegwa, P.A., S.O. Akowe, P.M. Otene and J.M. Ikyule, 2014. An overview on intuitionistic fuzzy sets. Intl. J. Sci. Technol. Res., 3: 142-145.

Elsawy, A. and W. Osamy, 2016. Genetic based algorithm for base-station position determination in wireless sensor network. Intl. J. Adv. Comput. Technol., 8: 16-27.

Ford, V., A. Siraj and M.A. Rahman, 2017. Secure and efficient protection of consumer privacy in advanced metering infrastructure supporting fine-grained data analysis. J. Comput. Syst. Sci., 83: 84-100.

Garg, H., 2017. Generalized interaction aggregation operators in intuitionistic fuzzy multiplicative preference environment and their application to multicriteria decision-making. Appl. Intell., 1: 1-17.

Hanafy, I.M., A.A. Salama, M. Abdelfattah and Y. Wazery, 2012. Security in Manet based on Pki using fuzzy function. IOSR. J. Comput. Eng., 6: 54-60.

Hanafy, I.M., A.A. Salama, M. Abdelfattah and Y.M. Wazery, 2013. AIS model for botnet detection in Manet using fuzzy function. Intl. J. Comput. Netw. Wirel. Mobile Commun., 3: 95-102.

Houssein, E.H. and A.A. Ismaeel, 2015. Ant colony optimization based hybrid routing protocol for MANETs. J. Emerging Trends Comput. Inf. Sci., 6: 622-629.

Praveena, A. and S. Smys, 2016. Efficient cryptographic approach for data security in wireless sensor networks using MES VU. Proceedings of the 10th International Conference on Intelligent Systems and Control (ISCO'16), January 7-8, 2016, IEEE, Coimbatore, India, ISBN: 978-1-4673-7808-6, pp: 1-6.

Sri, P.U., S. Sravani and G. Priyanka, 2017. Security protocols for wireless sensor networks. Intl. J. Eng. Comput. Sci., 6: 20289-20292.

Stallings, W. and M.P. Tahiliani, 2014. Cryptography and Network Security: Principles and Practice. Pearson, London, UK.,.

Wazery, Y.M. and M. Abd-ELfattah, 2017. PKI session key distribution in WSN using fuzzy rule based system. Intl. J. Sig. Syst. Control Eng. Appl., 10: 48-60.