https://doi.org/10.20943/01201702.5866

Extended SPINS Framework for Security Wireless Sensor Network

Khalid M. Abdullah[1], Essam H. Houssein[2], and Hala H. Zayed[1]

**[1]Computer Science Department, Faculty of Computers and Informatics,
Benha University, Egypt**

**[2] Computer Science Department, Faculty of Computers and Information,
Minia University, Egypt**

Abstract

Security in Wireless Sensor Networks (WSNs) plays an important role in the node communication. The significant growth is existed for developing the wireless sensor network applications. The key features of Wireless sensor networks are low power, low memory, and low-energy. Due to this various facts, the existing security algorithms are not appropriate for current applications. Many papers have been developed in this work. We describe the SPINS protocol of security for WNSs. In this work, the Sensor Protocols for Information via Negotiation (SPINS) is explained. It is a framework that implements the overall security in WSN with the use of Sensor Network Encryption Protocol (SNEP) and the "micro" version of the Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol (μTESLA) protocol. The previous research papers reveal that RC5 encryption algorithm done employed by SNEP for WSN. The proposed technique can defend against most well-recognized attacks in sensor networks, and achieve better computation and communication performance due to the more efficient algorithms based on AES and RSA.

Keywords: *security, TEA, AES, Wireless sensor network, SNEP, SPINS.*

1. Introduction

The rapid emergence of Wireless Sensor Networks (WSN) has come due to different reasons. Of the most important reasons is battlefield communications all to the communications in unreachable areas. However, with that came one drastic problem that faces WSNs which is their lack of security. In such way, WSNs are prone to be attacked; thus, the need of security solutions is necessary [1]. It is worth noting, however, that a large number of WSN architectures have been proposed and a key distribution solution that is well suited to one architecture is likely not to be the best for another, as different network architectures exhibit different communication patterns. Recent works show that there is a rapid advancement in a technology known as Micro-Electro-Mechanical Systems (MEMS).

Other developments have been made in the fields of digital electronics and wireless communications as seen in the recent years. Those advancements have allowed the decrease of costs and power while they increased the multi-functionality of sensor nodes. Sensor nodes come in small sizes and communicate through short distances. They consist of sensing, data processing, and communication mechanisms. Sensor networks represent major improvements over traditional sensors, which are deployed in the following two ways [2]:

- They can be placed far from the phenomenon and the larger they are, the more complex the technique they use is to differentiate between the targets of the environment.
- A number of sensors can perform only sensing and they can be deployed. However, the positions of those sensors and communications topology should be thoughtfully engineered. They transmit time series of the sensed phenomenon to the central nodes where computations are performed where the data are fused.

There are a lot of sensor nodes in a sensor network which is deployed within or closely around the phenomenon. The placement of a sensor network doesn't require planning ahead, which allows it to be randomly placed within the inaccessible areas. This makes the sensor networks and algorithms self-organize their abilities. One of the unique features in sensor networks is that they can use cooperative efforts with an on-board processor which means that instead of sending data while it's in its raw state, they use the processing ability to locally carry out simple computer equations and transmit the necessary data only [3-6].

However, another challenge presents itself to WSNs which is routing. This is due to several aspects that identify them from other modern communication and wireless ad hoc networks. It is almost impossible to design a universal addressing scheme for sensor node deployment, for that reason, classical IP-based protocols cannot be applied to sensor networks. Most of the applications of sensor networks need the flow of sensed data from multiple resources to a specific sink. The created data traffic has major dissolutions in it due to multiple sensors having the ability to create the same data with the area of the phenomenon. Thus, in order to improve the energy and bandwidth usage, the repetition in the routing protocols should be exploited [7].

Routing security in WSNs should be a point of focus. While many proposals for routing protocols in WSNs aimed to upgrade the limited capabilities of the nodes and the application specific in nature, they pay so little focus to the security. The design of a secure routing protocol stands out with great importance because the defender has the weaknesses in insecure wireless communications and limited node capabilities along with other security threats. Adversaries can use machines more powerful and with higher energy and longer reach to attack the network. In more conventional networks, security of the routing is necessary to guarantee message availability. In that sense, the message integrity, authenticity, and confidentiality are all handled at a higher level layer by end-to-end security mechanisms like SSH and SSL. The security in end-to-end can be done in a more orthodox network due to the unnecessity for intermediate routers to have access to the information within the messages. However, in sensor networks, the in-network processing makes end-to-end security mechanisms harder to deploy because intermediate nodes need direct access to the content of the messages. Link layer security mechanisms can help mediate some of the resulting weaknesses. However, this is not enough but it is not enough. More from routing protocols should be demanded [8].

Sensor nodes are deployed richly in order to sense and collect information. People interact with the environment differently due to the impact of WSNs. WSNs were originally inspired by military applications such as battlefield surveillance. However, today, WSNs are used in a lot of other applications industrial and consumer-wise. Wireless nodes of the sensor are used from different types of applications to collect the information like tracking facilities of critical, monitoring habitats of animals, and surveillance. They are discrete moneys spatially forced sensing devices taking small batteries and are normally capable of measuring human phenomena

In another work, the researchers have proposed a set of the security framework, SPINS, for WSNs. The SPINS is a combination two this: SNEP (Sensor Network Encryption Protocol) and μTESLA (the "micro" form of the Timed, Efficient, Flowing, Loss-tolerant Confirmation Protocol). SNEP provides data secrecy, data freshness and two-party confirmation where μTESLA provides authenticated broadcast for bleakly resource-constrained backgrounds. SPINS is a model that propose possible ideas for both base location to node and node to node announcement. So far, SPINS environment is one of the best definite and known WSN protection model. The main drawback of SPINS is that it uses RC5 algorithm for encryption that expansions the energy cost of SNEP. Present work shows that RC5 encryption algorithm is used in SNEP [9].

WSNs usually consist of thousands of sensor nodes and are deployed for different applications, including military sensing and tracking, environment monitoring, patient monitoring, etc. When sensor networks are deployed in a hostile environment, security becomes extremely important [10]. A wireless sensor network contains a big number of sensor nodes and at lowest one base station. The sensor node is a self-governing small device that comprises of four parts: sensing, processing, announcement, and power supply. These sensors are used to collect the information from the place and pass it on to the base place. A base station offers a relationship to the wired world where the collected data is managed, analyzed and presented to useful applications. Thus by embedding processing and announcement within the physical world, Wireless Sensor Network (WSN) can be used as a device to bridge real and virtual environment [11].

A Wireless sensor network (WSN) are machines equipped with a processor, a radio interface, memory, an analog-to-digital converter, sensors, and a power supply. The processor offers the mote administration functions and achieves data processing. The sensors attached to the mote are capable of sensing temperature, humidity, light, etc. Due to bandwidth and power constraints, motes primarily support low data units with limited computational control and a limited sensing rate. Memory is used to store programs (instructions executed by the processor) and data (raw and processed sensor measurements). Motes are equipped with a low-rate (10–100 kbps) and short-range (less than 100 m) wireless radio, e.g., IEEE 802.15.4 radio to connect among themselves. Since radio communication consumes most of the power, the radio must incorporate energy-efficient communication schemes. The power source commonly used is rechargeable batteries. For example, motes may be equipped with active power harvesting methods, such as stellar cells, so they may be left unattended for years [12].

The rest of this paper is organized as follows. In Section 2, we discuss WSN protocols. We present related work in Section 3. SNEP encryption method for SPINS framework is covered in Section 4. In Section 5, we describe the proposed technique (Extended SPINS Framework). Evaluations of proposed and existing encryption methods for SPINS framework are introduced in Section 6. Finally, Section 7 concludes the paper.

2. WSN Protocols

2.1 Security Vulnerabilities in WSNs

WSNs have many weaknesses when it comes to attacks. There are three major types of attacks [13]: Attacks on Secrecy and Authentication; Attacks on Network Availability; and Stealthy Attacks against Service Integrity.

- Attacks on Secrecy and Authentication include eavesdropping, packet reply attacks and modification of packets.
- Attacks on Network Availability include Denial-of-Service (DoS) attacks.
- Stealthy Attacks aim to accept incorrect data values as an example.

2.2 Hierarchical Protocols

A number of research projects have explored hierarchical clustering in WSNs [14]. Clustering is an energy-efficient communication protocol. It is used through the sensors to report sensed data back to the sink.

2.3 Application domains and deployments

WSNs have been accepted in a big number of diverse application domains [1]. It is proposed that in future everyday objects will be fixed with sensors to make them smart. Smart objects canister explore their location, communicate with other smart objects, and interact with humans. A taxonomy of WSN applications is shown in Fig. 2. In general, WSN applications can be of two types: monitoring and tracking. The primary application domains of WSNs include military and crime prevention, environment, health Body Area Networks (BAN), industry and agriculture, and development and organization.

2.4 Security Requirements

- Data Confidentiality: which means keeping information secret from unauthorized parties. The standard approach for keeping sensitive data secret is encrypting the data with a secret key that only intended receivers possess, to achieve confidentiality.
- Data Authentication: which stops unauthorized parties from the contribution in the network and legitimate nodes must be able to detect messages from unauthorized nodes and reject them.
- Data Integrity: which ensures the receiver that the received data is not changed in transit by a challenger.
- Data Freshness: which suggests that the data is recent, and it ensures that a challenger has not repeated old message [15].

3. Related Work

SPINS (Security Protocols for Sensor Networks) SPINS [16] is optimized for resource-constrained environments and wireless communication. SPINS has SNEP (Secure Network Encryption Protocol) and μTESLA (the "micro" version of the Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol).

Sensor protocol for information via negotiation (SPIN), is a protocol that broadcast all the information to every node in the network. Every node has related data with the neighboring node. The protocol distributes information to all nodes while the user doesn't require exchanging data between nodes. SPIN is a 3-stage protocol which uses three messages such as ADV, REQ & DATA. ADV is advertising new data, REQ is requested for data. DATA is the message itself. When a node wants to share data it broadcast an ADV message containing data. If the neighbor node is interested in receiving the data then it sends an REQ message back to the node for data transmission & DATA is send to the node [17]. Then the neighboring nodes repeat this process with its neighbors and the whole sensor area network will receive a copy of the data as shown in Figure 1.

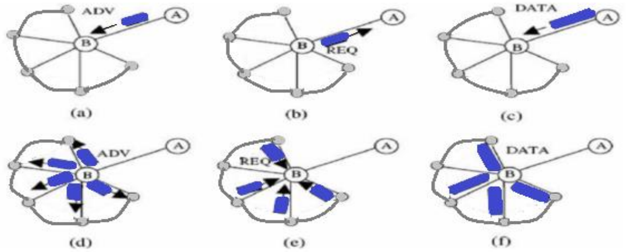

Fig. 1 (a) node A starts by advertising its data to node B, (b) node B responds by sending a request to node A, (c) after receiving the requested data, (d) node B then sends out advertisements to its neighbors, (e)(f)who in turn send requests back to B

SPIN is a number of protocols for WSNs. The goal of those protocols is to avoid the drawbacks of overflowing protocols by utilizing data negotiation and resource adaptive algorithms [18]. SPIN is designed based on two basic ideas; to operate powerfully and to save energy by sending meta-data (i.e., sending data about sensor data instead of sending the whole data that sensor nodes already have or need to obtain), and nodes in a network must be aware of changes in their energy resources and adapt to these changes to extend the operating lifetime of the system. SPIN has three types of messages, namely, ADV, REQ, and DATA. ADV: when a node has data to send, it advertises via broadcasting this message containing meta-data (i.e., descriptor) to all nodes in the network.

SNEP provides data confidentiality, two-party data authentication, data integrity, and data freshness. Before encrypting the message with a linking encryption function, the sender precedes the message with a random bit string. This prevents the attacker from inferring the plaintext of encrypted messages if it knows plaintext-ciphertext pairs encrypted with the same key. In order to avoid additional transmission overhead of these extra bits, SNEP uses a shared counter between the sender and the receiver for the block cipher in counter mode (CTR). The communicating parties share the counter and increment it after each block.

The encrypted data has the following format: E= {D} <Kencr, C>, where D is the data, the encryption key is Kencr, and the counter is C. The MAC is M=MAC (Kmac, C||E). The keys Kencr and Kmac are derived from the master secret key K. The complete message that A sends to B.

SNEP properties [19]:

- Semantic security: Because the counter rate is incremented after every message, the same message is encrypted differently every time. The counter value is passably long enough to not ever repeat within the time of the node
.

- Data authentication: If the MAC verifies correctly, a receiver differentiates that the message created from the required sender.
- Replay protection: The counter value in the MAC prevents a repeat of old messages. Note that if the counter were not existent in the MAC, an opponent could easily repeat messages.
- Weak freshness. If the message authenticates correctly, a receiver knows that the message need have been sent after the preceding message it received correctly. This applies a message ordering and yields weak freshness.

4. SNEP Encryption Method for SPINS

There are several pros for the SNEP system. First, it adds only 8 bytes to each message which builds low communication overhead. Second, it also uses counters but avoids transmission over keeping the state at both finish points. Third, it achieves semantic protection which avoids eavesdroppers from inferring the message contented from the encrypted letter. The RC5 algorithm is used to encryption. RC5 [2] was proposed by Professor Ronald Rivest of MIT and first published in December 1994. It has to change key size (0 to 2040 bits), block size (32, 64 or 128-bits) and no of rounds (0 to 255). Due to a trade-off between security and effectiveness, RC5-64/12/16 is suggested. In which, 64 bits word length, 12 no. used of rounds and 16 bytes key. The encryption and decryption are done in little lines of code but the key plan is more complicated. It is shown in Figure 2

4.1 RC5 Key Expansion Method

The key expansion algorithm is illustrated below, input to RC5 Plaintext contains 2 w-bit words which are denoted A and B [20].

Key k is User's expanded to fill the key table S which resembles a matrix of t=2(r+1) random binary words determined by k. where r Represent a number of rounds. The key expansion function takes a certain amount of "one-wayness": it is not so easy to determine k from S. Key expansion is achieved using the magic numbers Pw and Qw. Where w is word length. Magic constants are computed as follows:

$$Pw = Odd ((e - 2)2w)$$

$$Qw = Odd ((\phi - 1)2w)$$

Where, e = 3.11828182 (base of natural logarithms), ϕ = 2.01803398 (golden ratio), Odd(x) is the odd inter nearest to x.

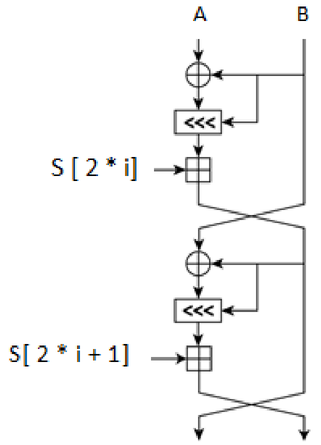

Fig. 2: Two half-rounds of the RC5 block cipher

Step 1: the secret key K[0, 1, … b-1] is copied into an array L[0, … c-1] of c = ceiling(b/u) words, where, u = w/8 is number of bytes/word. Any empty byte locations of L are zeroed.

Step 2: matrix S is primed to a particular fixed pseudo-random bit pattern, using magic numbers Pw and Qw.

```
    $S [0] = Pw;$
    For $i=1$ to $t-1$ do
    $S[i] = S[i-1] + Qw;$
```

Step 3: mixing of user's secret key executed in three passes over the matrix of S and L as follows:

```
    $i=j=0;$
    $A=B=0;$
    do $3*$ max $(t, c)$ times:
    $A = S[i] = (S[i] + A + B) <<< 3;$
    $B = L[j] = (L[j] + A + B) <<< (A + B);$
    $i = (i + 1) \bmod(t);$
    $j = (j+1) \bmod(c);$
```

5. Extended SPINS Framework

5.1 Two-phase Hybrid Cryptography Algorithm

Two-phase Hybrid Cryptography Algorithm (THCA) is a method that combines the aspects of symmetric and asymmetric techniques in performing a two parallel phase. They aim to avoid the obscurities in available technique through the realization of high-security levels without increasing the execution time.

In the encryption, the plaintext is divided into n blocks, Bi. Each block comprises of 64 bits. If n is not an integer number and has a fraction, DRSM algorithm uses padding with null for the last block to be 64 bits. The encryption process is divided into two phases.

In Phase I, plain text is divided into n/2 blocks.

First, part is encrypted using (DES) algorithm and Second part are used encrypted (RSA) algorithms.

Phase II is performed after of Phase I by combining the encrypted blocks together.

To increase the security, the output of Phase II the blocks are encrypted by using (AES) algorithm.

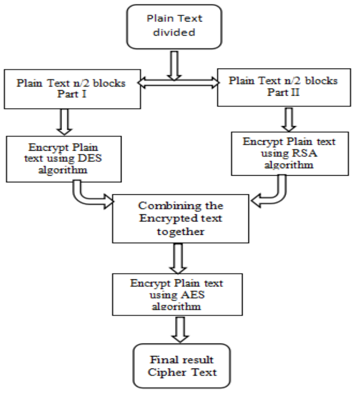
Plain Text divided
Plain Text n/2 blocks Part I
Plain Text n/2 blocks Part II
Encrypt Plain text using DES algorithm
Encrypt Plain text using RSA algorithm
Combining the Encrypted text together
Encrypt Plain text using AES algorithm
Final result Cipher Text

Fig. 3: Two-phase Hybrid Cryptography Algorithm ESPINS

5.2 Date Encryption Standard (DES):

The DES (Data Encryption Standard) algorithm is the most widely used encryption algorithm in the world. It is a block cipher that uses the same binary key both to encrypt and decrypt data blocks. It is called a symmetric key cipher. DES operates on 64-bit "plaintext" data blocks, processing them under the control of a 56-bit key to produce 64 bits of encrypted cipher-text. Similarly, the DES decryption process operates on a 64-bit cipher-text block using the same 56-bit key to produce the original 64-bit plaintext block. It uses a series of operations to encrypt a data block. These primitives are used to inverse the encryption operation as shown in Figure 4. Horst Feistel defined a variety of substitution and permutation primitives which are iteratively applied to data blocks for a specified number of times. Each set of primitive operations is called a "round," and the DES algorithm uses 16 rounds to ensure that the data are adequately twisted to meet the security goals. The secret key is used to control the operation of the DES algorithm. Each key contains 56-bits of information, selected by each user to make the results of the encryption process secret to that user. Any of about 1016 keys could be used by the DES, and an attacker trying to "crack" a DES encrypted message by "key exhaustion" (trying every key) must, on average, try half of the total possible keys before succeeding [21].

The predominant weakness of DES is its 56-bit key which, more than sufficient for the time period in which it was developed, has become insufficient to protect against the unbreakable force attack by modern computers [22].

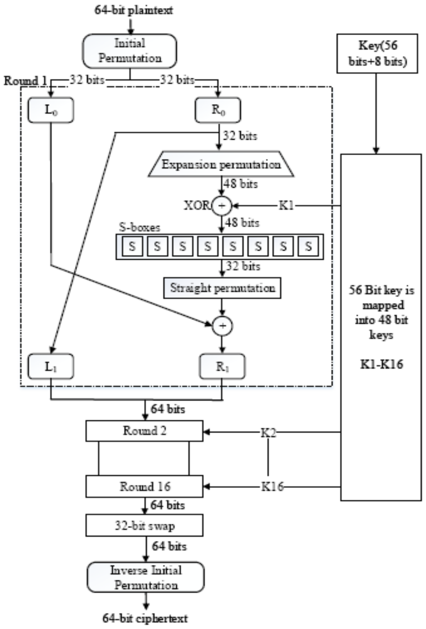

Fig. 4: General Depiction of DES [23]

5.3 Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES), also known in Rijndael [24], is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

The Advanced Encryption Standard (AES) algorithms is used to encrypt (encipher) and decrypt, (decipher), information.as shown in Figure 5. The Key Development generates a Key Agenda that is used in Cipher and Converse Cipher procedures. Cipher and Inverse Cipher are composed of a specific number of rounds. The number of rounds to be performed through the application of the algorithm depends on the key length. The four changes used in every round are [25].

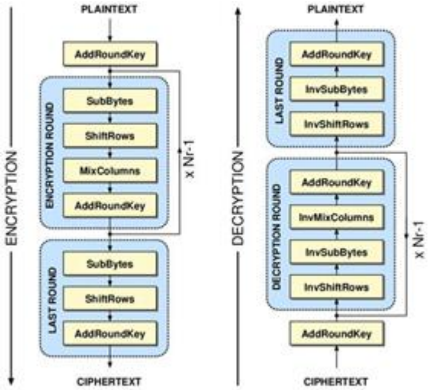

Fig. 5: AES Encryption and Decryption way

Sub Bytes

The sub-byte transform is shown in Figure 4. In AES algorithm the function of the sub-byte is only nonlinear function and that operates independently on each byte of the state using substitution table (S-box). It substitutes all bytes of the state array using an LUT which is a 16x16 matrix of bytes, often called S-box. In AES hardware implementation, S-box design contributes a major role in optimization two approaches for S-box design. Design a multiplicative inversion and affine transformation separately or Construct a logic circuit defining the input and output of the S-box function [26].

Shift Rows

The rows in the general array are replaced. The byte in the first row is not changed but second, the third and fourth row is changed by shifted left by one byte.

Add Round Key

For each 128-bit round, the key is divided into 16 bytes as of data block. A round key is added to the State matrix by bitwise Exclusive-OR (XOR) process Key is used as a primary set of bytes in each row and the rest of the bytes are created from the key iteratively.

Mix Columns

Mix Columns is a linear change and is completed on the state matrix by column by column. Every one of the changed bytes is a linear concoction of the state array.

The key expansion

The term is used to describe the process of generating all Round Keys from the original input key. The original round key will be the original key in case of encryption. The whole operation is shown in Figure 6. The key expansion term is used to describe the operation of generating all Round Keys from the original Input key.

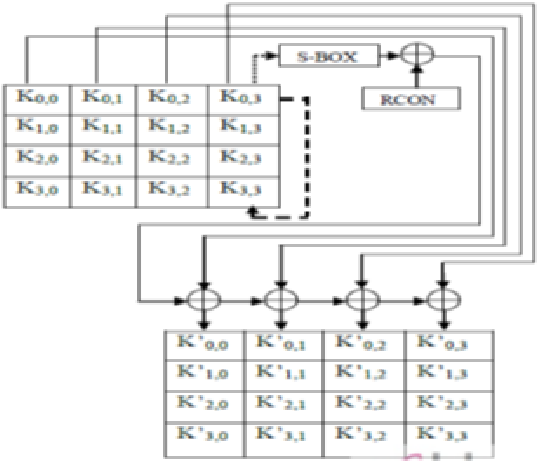

Fig. 6: Key Expansion

5.4 Rivest-Shamir-Adleman (RSA):

Ron Rivest, Adi Shamir, and Leonard Adleman invented the RSA which is one of the most popular public key cryptosystems for key exchange or digital signatures. It uses an adjustable size encryption block and a size key that can change. The public key is an asymmetric cryptosystem that is based on several theories where it manages to use two prime numbers to create both the public and the private keys. They are used for both encryption and decryption where the sender encrypts the message and the receiver decrypts it with the use of those keys. However, the receiver only uses their own private key for the decryption while the sender uses the key for transmitting the message as well [27].

The public key in this cryptosystem comprises of the value n, which is called the modulus, and the value e, which is called the public exponent. The private key comprises of the modulus n and the value d, which is called the private exponent.

An RSA public-key / private-key pair can be generated by the following steps:
1. Create a pair of large, random primes p and q.
2. Compute the modulus n as n = pq.
3. Select an odd public exponent e between 3 and n-1 that is relatively prime to p-1 and q-1.
4. Compute the private exponent d from e, p, and q. (See below.)
5. Output (n, e) as the public key and (n, d) as the private key.
The encryption operation in the RSA cryptosystem is exponentiation to the eth power modulo n:

$$c = Encrypt(m) = m^{e} \bmod n \qquad (1)$$

The input m is the message; the output c is the resulting ciphertext. In practice, the message m is typically some kind of suitably formatted key to be shared. The actual message is encrypted with the shared key using a traditional encryption algorithm. This structure makes it possible to encrypt a message of any length with only one exponentiation. The decryption operation is exponentiation to the dth power modulo n:

$$m = Decrypt(c) = c^{d} \bmod n \qquad (2)$$

6. Evaluation of Proposed Technique

6.1 The size of the cipher text

Table 1 describes the output of the encryption process. It shows the size of the cipher text in bytes. It is shown that the protocol proposed by SPIN and SPEN had the least favorable results.

Table 1 Size of Cipher Text (bytes)

Size of plain text (byte)	SPEN	SPINS	ESPINS
609	673	846	648
25615	25645	35142	25647
35080	35192	48226	35116

6.2 Throughput

The time of encryption is used to computing the throughput of an encryption model. It indicates the encryption speed. The throughput of the encryption model is computing as the following reference [28]:

$$Throughput = T_{p}(Bytes) / E_{t}(Sec) \qquad (3)$$

Where Tp is the total plain text is measured in the (bytes) and Et is the time of encryption (second). Figure 7 shows the throughput of *ESPINS* proved to be better in comparison to the existing protocols for the different sizes of the plain-text.

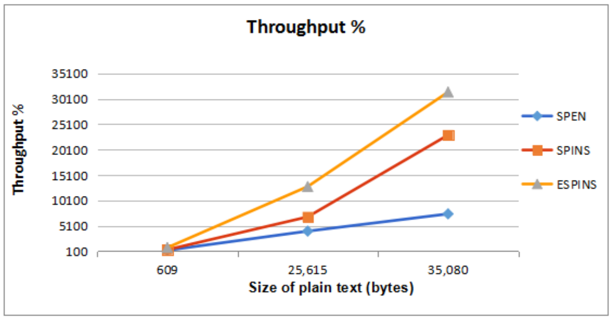

Fig. 7: shows the throughput of ESPINS comparison SPEN and SPINS

6.3 Simulation Results of WSN

So as to show the measured results of the proposed protocol, it is tested as the protection protocol in WSN. The imitation is done using the network simulator ns2. It contains twenty nodes. The nodes are located randomly in the network. In the assumed scenario, Node “0” wants to transport the data to Node “18”.

Information about the other nodes within one WSN should be available to all the existing nodes. It is initially transmitted in small packets, where each packet contains information about the source address. If the transitional node receives a packet, it automatically sends it to the neighboring node and so on. When the packet reaches the final node then all the addresses within the packet are checked and it transmits a reply to the source node. The packet size witnesses a gradual increase.
After the transmission, each node will identify the location of all the other sensor nodes within the network and so the communication would be done from one node to the other. However, when the links between the nodes fail, some cases propagate over the link between two nodes.

The hybrid algorithm (DES) with (RSA) is used to delete some packets because of increase to the time of execution (time out) as shown in Figure 9. When such nervous packets are dropped, the link will not be used for time of a certain. As a result, the network will be then using a substitute path as the nodes are having the information about the other nodes in this network for validation.

Figure 8 shows the execution time of ESPINS compared with protocols (SPEN and SPINS). It is shown that the ESPINS achieves the least time of execution. When the time of execution increases, it leads to increase of fallen packets.

Figure 9 shows the rate of fallen packets. It is shown that the ESPINS achieves the least rate of packet fallen compared with protocols.

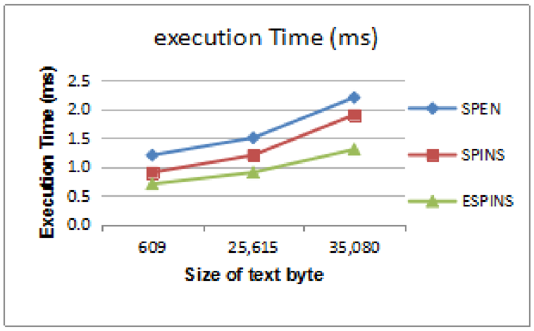

Fig. 8: The execution time of ESPINS with the comparison of the other protocols

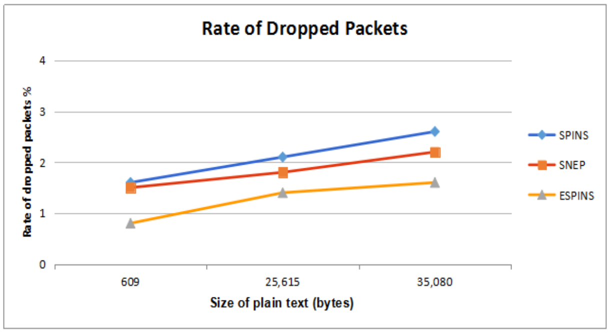

Fig. 9: The rate of fallen packets of ESPINS and the existing protocols

7. Conclusions

The proposed hybrid algorithm combines characteristics of the symmetric encryption of both AES and DES and the asymmetric encryption of the RSA. That provides an easy and a fast way of securing information transmission. Overall, the hybrid method proposed has several good advantages such as the simplicity principle and high-security standards due to the multi-level encryption used. It also decreases the rate of dropped packets. The proposed work was compared to two previously proposed protocols and it proved to come up with better results in terms of security, efficiency, and speed.

References

[1] L. B. Oliveira, A. Ferreira, M. A. Vilaca, H. Chi Wong, M. Bern, R. Dahab, A. A.F. Loureiro, “SecLEACH—On the security of clustered sensor networks”, Elsevier, Signal Processing 87, 2882–2895, 2007.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “Wireless sensor networks: a survey”, Elsevier, Computer Networks 38, 393–422, 2002.

[3] Radwan, A., Mahmoud, T. M., & Houssein, E. H. (2011). Performance measurement of some mobile ad hoc network routing protocols (IJCSI), 8(1), 107-112.

[4] Sadeghi, M., Khosravi, F., Atefi, K., & Barati, M. (2012). Security analysis of routing protocols in wireless sensor networks. International Journal of Computer Science Issues (IJCSI), 9(1), 465-472.

[5] Sookhak, M., Karimi, R., Ithnin, N., Haghparast, M., & ISnin, I. F. (2011). Secure Geographic Routing Protocols: Issues and Approaches. International Journal of Computer Science Issues (IJCSI), 8(5).

[6] Houssein, E. H., & Ismaeel, A. A. (2015). Ant Colony Optimization based Hybrid Routing Protocol for MANETs. Journal of Emerging Trends in Computing and Information Sciences, 6(11).

[7] K. Akkaya, M. Younis, “A survey on routing protocols for wireless sensor networks”, Elsevier, Ad Hoc Networks 3, 325–349, 2005.

[8] C. Karlof, D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures”, Elsevier, Ad Hoc Networks 1, 293–315, 2003.

[9] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. D. Tygar, “SPINS: Security Protocols for Sensor Networks”, Wireless Networks 8, 521.534, 2002.

[10] J.P. Walters, Z. Liang, W. Shi, and V. Chaudhary,“Wireless Sensor Network Security: A Survey,” Technical Report MIST-TR-2005-007, July 2005

[11] S. K. Gupta, P. Sinhha,” Overview of Wireless Sensor Network: A Survey”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 1, Jan. 2014.

[12] P. Rawat, K. D. Singh, H. Chaouchi, J. M. Bonnin,” Wireless sensor networks: a survey on recent developments and potential synergies”, J Supercomput 68:1–48, 2014.

[13] J. D. Wheeler, M. R. Needham, “a Tiny Encryption Algorithm”, Computer Laboratory, Cambridge University, England. Nov. 1994.

[14] H. Kaur,“Survey on Routing Protocols for Wireless Sensor Networks“, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5, Issue 4, April 2015

[15] J. Yick, B. Mukherjee, D. Ghosal, “Wireless sensor network survey”, Computer Networks 52 2292–2330, 2008.

[16] D. Xiao ,W. Meijuan Wei, Y. Zhou, “Secure-SPIN: Secure Sensor Protocol for Information via Negotiation for Wireless Sensor Networks”, Industrial Electronics and Applications, 1st IEEE Conference, 2006.

[17] R. K. Magal, M. Revathy, “A Survey on Wireless Sensor Network Protocols”, IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 6, Aug. 2014.

[18] P. J. Chauhan, K. M. Pattan, “A Survey Security Protocol for Wireless Sensor Network“, International Journal of Modern Trends in Engineering and Research (IJMTER) Volume 01, Issue 06, Dec. 2014.

[19] A. Pandey, R.C. Tripathi, “A Survey on Wireless Sensor Networks Security”, International Journal of Computer Applications (0975 – 8887), Volume 3 – No.2, June 2010.

[20] R. Rivest, “The RC5 Encryption Algorithm”, in Fast Software Encryption of Lecture Notes in Computer Science, Preneel, B., Ed., Springer-Verlag, vol. 1008, pp: 86-96, 1995.

[21] J. Orlin Grabbe 2006, the DES Algorithm Illustrated, Laissez Faire City Times, Vol 2, No. 28.

[22] R. Stephen Preissig, 2000, Data Encryption Standard (DES) Implementation on the TMS320C6000, Application Report SPRA702 November 2000

[23] Singh, G., Supriya, 2013. A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. Int. J. Comput. Appl. 67 (19), 33–38

[24] K. Avi, K. Mayes, “AES: The Advanced Encryption Standard”, Lecture Notes on Computer and Network Security, Purdue University (2013).

[25] V. Saini, P. Bangar, H. S. Chauhan, “Study and Literature Survey of Advanced Encryption Algorithm for Wireless Application“, International, Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume2, Issue-6, Apr. 2014.

[26] J. V. Shiral, R. C. Deshmukh, J. S. Zade, A. Potnurwar, “an Approach to Rijndael Algorithm”, Journal of The International Association of Advanced Technology and Science, Vol. 16, ISSN-5563-168, Feb. 2015.

[27] Tom Davis, 2003, RSA Encryption, 2003.

[28] S. Al-alak, Z. Ahmed, A. Abdullah and S. Subramiam," AES and ECC Mixed for ZigBee Wireless Sensor Security", World Academy of Science, Engineering and Technology, 2011.