International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

Fast Encryption for Multi-Hop Wireless Sensor Networks using Gaussian Transposition Cipher

Amarthaluri Thirupathaiah, IB Venkateswarlu

Abstract*: *Wireless sensor network research enriched with diverse applications from industry to daily life. Widespread of sensor-based applications mandated for user authentication and secure communication. However, sensor nodes are limited energy and resources and hence secure communication for sensor nodes became a challenging task. This paper presents a fast encryption scheme for secure communication in wireless sensor networks. The proposed scheme consists of three phases namely registration, network deployment, and data transmission. In this work, a Gaussian transposition cipher for the generation of strong key. This cipher uses Gaussian noise, modified rail fence cipher and transposition. Fast encryption has achieved using XOR-based encryption and hence the proposed scheme incurs low computational cost. The proposed scheme resistant to various security attacks.

Keywords* : *Wireless sensor network, Fast encryption scheme, Gaussian transposition cipher, Seeded rail fence.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) became an emerging field in sensor network research. WSN records rapid growth with diverse sensor-based applications including industrial internet of things [1], agriculture [2], video surveillance [3], and ground monitoring [4]. Traditional WSN equipped with several sensor nodes and sink nodes depending on the task as follows.

- Sensor node is the main sensing device that tackles actual sensing work. It is equipped with limited resources to cater to the need of a sensing environment. Sensor nodes are battery operated and hence efficient utilization of energy is the primary concern in WSNs.
- Sink node is a system that collects sensed data from various sensor nodes. It uses data aggregation to collect data and also performs various tasks based on the application.

In general, sink nodes are rich in resources while sensor nodes have limited resources [5]. Transmitter, receiver and sensing circuit are the major components of the sensor node. Every sensor node has capabilities of computation, identification, and communication. Wireless sensor network models can be classified into two types as Single-hop and Multi-hop depending on the number of hops. Sensor nodes directly send sensed data to the sink node without any intermediate node in single-hop model. On the other hand, sensor nodes send sensed data to the sink node through intermediate nodes in multi-hop model.

This intermediate node is also known as Cluster Head (CH) which performs a re-transmission of aggregated data to the sink node. Fig. 1 depicts a simple multi-hop WSN model consisting of 8 sensor nodes, 3 cluster heads, and a sink node. In this model, four-sensor nodes N1 to N4 are connected to cluster head CH1. Similarly, CH2 is connected with two nodes (N5 and N6) and CH3 is connected with two nodes (N7 and N8). Finally, all three cluster heads communicate with the sink node.

Nowadays, security has evolved as one of the important research tasks in WSN to accomplish confidential applications like Military, Health care, etc.. Thus, Wireless sensor network security became challenging tasks as wireless networks are more vulnerable than wired networks [?]. Most security mechanisms try to realize the triple security goals including confidentiality, integrity, and availability and are defined as follows.

- Confidentiality prevents unauthorized access or the revealing of the information in the process of data transfer.
- Integrity ensures that data are sent to the receptor in a correct, exact, and complete.
- Availability ensures that authorized users have confidence in a timely.

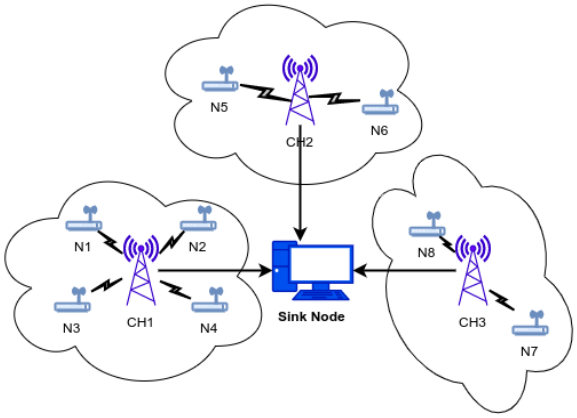

Fig. 1. Multi-hop wireless sensor network basic model

Revised Manuscript Received on March 30, 2020.
* Correspondence Author
Dr. Amarthaluri Thirupathaiah*, Department of CSE, St. Ann's College of Engineering and Technology, Chirala, India
IB Venkateswarlu, Department of CSE, St. Ann's College of Engineering and Technology, Chirala, India

Consider a multi-hop WSN intruder model in which intruder has compromised CH1 and CH3 as shown in Fig. 2. Then, the network is vulnerable to various security threats. As sensor nodes in WSNs broadcast their sensed data over the network and hence any intruder connected to the network can also have access to the data.

Thus, authentication plays a major role to restrict such un-authorized access. Some active intruders may send packets to the sink node or cluster head by changing the contents of packets or by creating fake data packets. This violates the confidentiality and integrity of WSN.

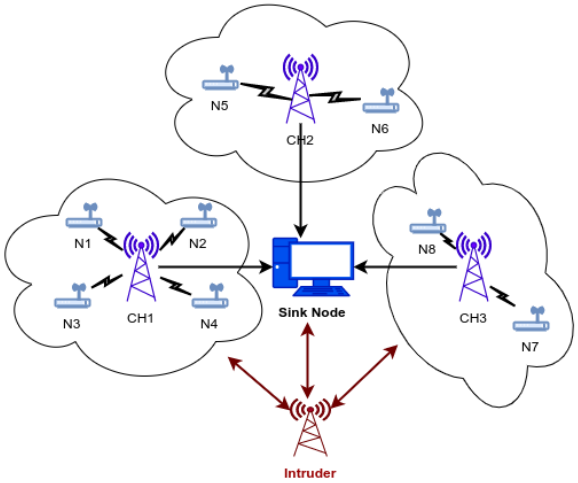

Fig. 2. Wireless sensor network intruder model

Moreover, processing of fake packets incurs fruitless energy consumption which leads to lower network lifetime. This motivated researchers to focus on various security aspects while implementation of wireless sensor network security schemes. However, user authentication and secure communication became the dominant task in wireless sensor network communication. It motivated us to implement an energy-efficiency secure wireless transmission scheme. The proposed scheme uses a Gaussian transposition cipher to generate a strong secure-key. each packet of the data. Then, XOR-based encryption is performed at sensor nodes for fast encryption.

II. RELATED WORK

Energy preserving secure measure has proposed by Aliady et al. [6] to detect wormhole attack. This measure is based on the network connectivity and applied to the ad-hoc-on-demand distance vector routing protocol. Simulation results on NS-3 stats that the detection accuracy is 100% when the wormhole tunnel is of four hops or more in length.

Nivedetha et al. [7] developed a kind of Fuzzy Fingerprint Biometric-based Key Security (FFBKS) scheme. The private key for the user is produced using utilization feature extraction. This key is sent to every sensor node. Another private key among sensor nodes is produced using pseudo-random number and user key. The adaptive possibilistic C-means clustering (APCMC) is utilized for clustering of nodes based on distance and identifier among nodes. The group key is produced based on a fuzzy membership function and is utilized for the estimation of security.

Deebak et al. [8] implemented a secure routing and monitoring protocol to discover and prevent the adversaries in the global sensor network. A secure hybrid routing protocol is selected to be built by inheriting the properties of both Multipath Optimized Link State Routing (OLSR) and Ad-hoc On-demand Multipath Distance Vector (AOMDV) protocols.

Biswas et al. [9] an energy-efficient lightweight encryption scheme based on pseudo-random bit sequence generated by elliptic curve operations. Initially, a random bit sequence generator produces a pseudo-random bit sequence. Then, given plain text is also converted into a bit sequence. Finally, XOR of plain bit sequence with pseudo-random bit sequence gives cipher-text.

III. GAUSSIAN TRANSPOSITION CIPHER

Key generation and management is the vital task in major encryption algorithms. Moreover, it is a more complicated and energy-consuming task in wireless sensor networks. The proposed encryption scheme utilizes a new Gaussian Trans-position Key (GK) for the encryption process. This section introduces key generation algorithms for Gaussian transposition key. The proposed key generation algorithm is a three-step process to produce a complex and secure key. Initially, given input data is passed to the Gaussian function for getting its Gaussian code. Then, the Gaussian code is concatenated with random code. Finally, Gaussian Transposition key is generated using transposition cipher. If the length of input data is N bytes, then its Gaussian code (GC) is computed using eq. 1.

$$GC(X) = \frac{e^{-(X-\mu)^2}}{2 * \sigma^i} \quad (1)$$

Where, μ is mean of the input data $X = \{x_1, x_2, ...x_N\}$, σ is standard deviation and i (≥ 1) is sigma deviation faction. More deviation fraction gives more complex Gaussian code. The proposed algorithm introduces Seeded Rail Fence (SRF) cipher which is a modified version of a rail fence cipher. Consider, pain text P as ”This is plain text.” and key (K) is 4, then the basic rail fence encryption is demonstrated in Fig. 3a.

The basic rail fence is not strong cipher as it uses a weak key (number of rails) which can be easily guessed by the crypt analysts. Thus, modified seeded rail fence cipher takes Seed (S) value in addition to the key (K) for the strengthening of the cipher. If P is plain text, K is key, and S is seed value then encrypted text E can be computed in two steps as follows.

1) SP = Perform circular (left or right) shift on P by S positions
2) E = Apply basic rail fence encryption process on SP

Fig. 3b visualizes the process of seeded rail fence encryption with S = 5. On the other hand, the decryption process is reverse to the encryption process which starts with basic rail fence decryption. Then, a circular shift of plain text by S positions is performed to get plain text.

Plain text (P) = "This_is_plain_text." Key (K) = 4

Encryption process:

T						s						n						.
	h				i		_				i		_				t	
		i		_				p		a				t		x		
			s						l						e			

Encrypted text (E) = Tsn.hiiti_patxsle

(a) Basic rail fence

Plain text (P) = "This_is_plain_text." Key (K) = 4 Seed (S) = 5
Shifted text (SP) = "is_plain_text.This_"
Encryption process:

i						i						t						_
	s				a		n				x		.				s	
		_		l				_		e				T		i		
			p						t						h			

Encrypted text (E) = iit_sanx.s_l_eTipth

(b) Seeded rail fence

Fig. 3. Encryption process of rail fence cipher

Timely changing of this seed value is recommenced to make basic rail fence as a complex cipher. Dynamic seeded rail fences becomes more complex for dictionary attacks.

Algorithm 1: Gaussian transposition cipher encoding (GKenc)

1) gc = compute GC(P) using eq. 1
2) rc = generate M byte random code
3) grc = gc||rc
4) GK = SRFenc(grc, K, S)

Algorithm 2: Gaussian transposition cipher decoding (GKdec)

1) grc = SRFdec(E, K, S)
2) < gc, rc >= grc
3) P = GC(gc)

Proposed cipher uses Gaussian code, random code and seeded rail fence cipher to generate strong key known as Gaussian transposition key (GK). If P is plain data consists of N bytes then required GK is generated using algorithm 1. Similarly, algorithm 2 gives the decoding process of given Gaussian transposition key.

IV. PROPOSED FAST ENCRYPTION SCHEME

In general, sensor nodes have limited computation and resource facilities. Thus, computational and energy-efficient algorithms are highly recommended for sensor nodes. However, the sink node has rich computation and resource facilities. In general, the strength of encryption algorithms depends on complexity of encryption process and strength of key used for encryption. Complex encryption provides high security and also incurs high computational cost as well as energy especially in wireless sensor networks. Thus, the proposed encryption scheme uses a simple XOR operation for both encryption and decryption. We have utilized a more secure key namely Gaussian transposition key to increase the security of encryption algorithm. The proposed fast encryption scheme for wireless sensor networks consists of three phases namely registration, network deployment, and data transmission. A detailed process of each phase is as follows.

A. Node registration phase

Each sensor of the network must register with a sink node. In this phase, the sink node needs to perform various operations as follows.

1) U_i = Rand()
2) $GK1_i$ = GKenc(U_i)
3) Store in NodeInfo(U, GK1, R, S, CHs)
4) Store < U_i , $GK1_i$, BSK > in sensor node

This phase creates *NodeInfo* table which is a collection of registered sensor node information at the sink node. Later, sink node uses this *NodeInfo* in the network deployment phase for node authentication and session creation. When a new sensor node arrives it should be registered at the sink node before communication. This registration process generates random unique user code (U_i) along with Gaussian transposition key for each user and stores in *NodeInfo*. Though, this process can be automated with a private communication channel, it is recommended to use an offline registration process for more security.

B. Network deployment phase

Data transmission will be initiated by the sensor nodes in this phase. Initially, each sensor node of the network sends ”Hello” message to the sink node.

This message consists of 16-bit node information (NO) used for authentication and 16- bit network information (NE) used for cluster selection. After receiving these messages from each sensor node, the sink node performs cluster head selection and then session initiation. After receiving "Hello" messages, designation of cluster heads will be communicated through acknowledgment message by the sink node. The sequence of steps need to be followed in this phase are as follows.

At sensor node:

1) Initially, i^{th} sensor node generates 16-bit random code R_i using any random function. Then, node unique code and random code are encrypted with Gaussian transposition key ($GK1_i$) and U_i respectively.
   - $R_i = Rand()$
   - $U_i = U_i \oplus GK1_i$
   - $R_i = R_i \oplus U_i$
2) Sensor node computes its normalized distance of node from sink node (D_i), normalized residual energy of node (E_i) and normalized angle of orientation with respect to sink node (A_i). Then, it creates "Hello" message (HM) of 64-bit using 32-bit node information (NO) and 32-bit network information (NE) as given below.
   - $NO_i = U_i \oplus R_i$
   - $NE_i = D_i \,||\, E_i \,||\, A_i$
   - $HM_i = NO_i \,||\, NE_i$
3) Later, each sensor node sends HM_i to sink node for authentication as well as session initiation.

At sink node:

1) Sink node receives HM of each sensor node.
2) It compares first 16-bits of each HM_i with U_i field of *NodeInfo*. If the node is authenticated then it will update *NodeInfo* and add the details to *MsgInfo* which is used for cluster head selection.
   - $U_i = U_i \oplus GK1_i$
   - *Check* U_i *, if unique code is invalid then discard*
   - *If unique code is valid then perform* $R_i = R_i \oplus U_i$
   - *Extract* D_i , E_i , A_i
   - *Update NodeInfo*$< R >$
   - *Add to MsgInfo*$< U, D, E, A >$
4) Then, sink node performs cluster head selection based on D_i , E_i and A_i . Any optimization algorithm can be used to select list of M cluster heads. One simple cluster head selection approach is as follows.
   - *Sort MsgInfo list such that D values are in ascending, E values are in descending and A values are in ascending.*
   - *Select top M senor nodes are selected as cluster heads*
5) After cluster head selection, update CHs status in *NodeInfo* $< CHs >$ and generates a random session key (S_i) per each session. Sink node encrypts M cluster head unique codes as follows.
   - $CH_i = CH_i \oplus BSK$
   - $S_i = Rand()$
   - *Store* S_i *in NodeInfo* $< S >$
   - $S_i = S_i \oplus GK1_i$
   - $CH_i' = CH_i \,||\, S_i$
5) Finally, sink node broadcasts encrypted cluster head list as acknowledgement (ACK) messages.
   - $ACK = CH_1' \,||\, CH_2' \,||...||\, CH_n'$
   - *Transmit ACK to sensor nodes*

At sensor node:

Each sensor node receive ACK, split into designated parts and then performs self updation as sensor node/cluster head as given below.

1) $< CH_i' , S_i >= ACK$
2) $CH_i = CH_i' \oplus BSK$
3) If $CH_i = U_i$ then
   - $S_i = S_i \oplus GK1_i$
   - *Store* S_i *in sensor node*
   - *Switch on transmitter and receiver*
4) Else discard
   - $GK2_i = GK1_i \oplus R_i$
   - *Switch off receiver*
   - *Switch on the transmitter*

If the node is sensor nodes, then it creates a second Gaussian key ($GK2_i$) and switches off the receiver. If the node is cluster head, then it stores session key (S_i) and switches on both transmitter and receiver.

C. Secure transmission phase

Once, authentication and session initiation process is over. Secure transmission can be started by sensor nodes through cluster heads.

At sensor node:

1) $D' = D \oplus GK2\ i$
2) $U_i' = U_i \oplus BSK$
3) $DM_i = U_i' \,||\, D'$

At cluster head:

1) Receives DM_i from sensor node
2) $DM_i' = DM_i \oplus S_i$
3) Transmit DM_i' to sink node

At sink node:

1) Receive DM_i' from CH_j
2) $DM_i = DM_i' \oplus S_j$
3) $< U_i' , D >= DM_i$
4) $U_i = U_i' \oplus BSK$
5) If U_i not found discord
6) Else
   - $GK2_i = GK1_i \oplus S_j$
   - $D = D' \oplus GK2_i$
   - *Process data D*

V. RESULTS AND DISCUSSION

Though there are several security mechanisms for wireless sensor networks, none of the mechanisms focuses on both authentication and secure communication. Thus, the proposed Fast Encryption Scheme (FES) addresses both authentication and fast encryption of sensor data. We have simulated the proposed scheme using Python and following assumption are considered while simulation.

- Location of all sensor nodes as well as cluster heads are static within the network area
- Congestion control over the broadcast is ignored i.e. channels are lossless

This section presents commonly used security evaluation approaches namely performance analysis and crypt-analysis. Computational cost and resource cost of proposed scheme are elaborated in performance analysis. Similarly, crypt-analysis exhibits strength of proposed Gaussian transposition key and encryption scheme to resist various attacks.

A. Performance analysis

The computational time of security scheme plays a vital role in wireless sensor networks. Complex algorithms incur high computation cost which causes more energy consumption at the sensor node. Thus, simple and less complex algorithms are recommended in WSN. This motivated to implement simple XOR-based encryption to achieve fast encryption. However, proposed scheme uses a strong Gaussian transposition key to increase cracking complexity for the intruder. Proposed fast encryption scheme uses three XOR operations at sensor node and only one XOR operation at cluster head for encryption. Sink node needs four XOR operations to decrypt sensor data. In general, XOR operations incur very little time when compared to any standard encryption. Encryption and decryption process of proposed fast encryption scheme on 32 bytes random message passed through cluster head is as follows.

Random message: (at sensor node)
3WEOT33HDHJFFRAH5AJW63TA7HQ7KASV
Encrypted message: (at sensor node)
%W".6=P6S$-/[ZG16,H$+\H:1:Q?[,7]
Encrypted message: (at cluster head)
r ubaq z hzc }a`h| vfv s``
Decrypted message: (at sink node)
FEJGXAA2ILGB0S0R2SDS8B25SQ8CROYR

Given message is encrypted by a sensor node using its GK2 and then sends to cluster head. Then, cluster head again encrypts the message using session key. Thus, the proposed encryption uses double encryption. It can be observed there is a possibility of some invisible characters in encrypted text as in an encrypted message at cluster head. This double encryption adds more complexity in encryption algorithm.

Biswas *et al.* has addressed fast encryption scheme like us namely simple lightweight encryption scheme. Table I compares execution time elapsed for the encryption of 32-byte data packets (encryption time of Biswas *et al.* is taken from the paper). It clearly shows proposed scheme takes few milliseconds while other methods takes around 6 seconds. However, proposed scheme key generation is comparatively time consuming process which takes 4.6515 milliseconds.

Table- I: Comparison of encryption time

Scheme	*Time (sec.)*
Biswas *et al.*	6.207000
Proposed	0.000109

Mean square error (MSE) is another important measure to evaluate strength of encrypted data. It represents error between plain text and encrypted text. If this error is more indicates that the encrypted text is more deviated from the plain text which is required for a strong encryption algorithm. Table II lists execution time and MSE of proposed FSE for various packet sizes including 32-bytes, 64-bytes, 128-bytes, 256-bytes, 512-bytes, and 1024-bytes. This table proves that proposed scheme achieves consistent MSE value irrespective of packet size.

Table- II: Encryption time MSE of proposed scheme

Scheme	*Time (msec.)*	*MSE*
32-bytes	0.1125	2478.4375
64-bytes	0.1619	2834.3750
128-bytes	0.2346	2676.3593
256-bytes	0.4270	2506.7421
512bytes	0.8006	2732.6464
1024-bytes	1.5514	2725.0058

However, proposed scheme utilizes additional memory to reduce computational time. Each sensor node stores U_i (2 bytes), R_i (2 bytes), GK1 (1024 bytes), and GK2 (2 bytes). Each cluster head stores U_i (2 bytes), R_i (2 bytes), GK1 (1024 bytes), and S_i (2 bytes). Thus, additional memory overhead at each sensor node as well as cluster head is 1030 bytes. Similarly, sink node stores complete details of network as *NodeInfo* and *MsgInfo*. As sink node is rich in resources and hence memory overhead at sink node can be ignored.

B. Crypt-analysis

Crypt-analysis is the process of checking strength of security schemes against various security attacks. Proposed security scheme is resistant to various security attacks as follows.

- **Quickly detection for unauthorized login**: proposed scheme uses XOR-based encryption which is a very fast encryption technique. Thus, authentication (detection of unauthorized node) in the proposed scheme is quite fast.
- **Sensor node anonymity**: every time sink node checks first 32-bits of data which is unique code of the sensor node. Thus, anonymous packets or sensor data is discarded in this checking process.
- **Resist replay attack**: proposed scheme uses random code with its unique code and hence its is highly resistant to replay attack.
- **Proper mutual authentication**: proposed scheme recommends offline sensor node registration and hence it is effective for mutual authentication.
- **Man-in-the-middle attack**: proposed scheme uses a random session key (S_i) and Gaussian transposition key ($GK1_i$). If decryption at sink code is invalid, it will discard packets.

Thus, this proves the resistance to man-in-the-middle attack. If intruder sends random data packets, sink node simply discard.

- **Dictionary attack**: It is popular key hacking attack by keeping dictionary of keys. Proposed method uses random code and Gaussian code for generation of key and that code is again XORed with random session key. Thus, it is more complex to crack our double randomized key using dictionary attack.

VI. CONCLUSIONS

This work presents a fast encryption scheme for wireless sensor networks. The proposed scheme achieves authentication and secure communication with three phases namely registration, network deployment, and secure transmission. Complex and secure Gaussian transposition key are generated using Gaussian code of node unique code and random code with seeded rail fence cipher. Proposed security scheme uses Gaussian transposition key and XOR-based encryption to achieve secure communication. The proposed security scheme performs three XOR operations and only one XOR operation at sensor node and cluster head respectively. Thus, computation cost of encryption is very less when compared with existing methods. The proposed scheme needs additional memory of 1030 bytes. Moreover, proposed scheme is resistant to various security attacks like sensor node anonymity, replay attack, dictionary attack, etc.. Our future work focuses on including congestion control along with security.

REFERENCES

1. X. Li, J. Peng, J. Niu, F. Wu, J. Liao, and K. R. Choo, "A robust and energy efficient authentication protocol for industrial internet of things," IEEE Internet of Things Journal, vol. 5, pp. 1606–1615, June 2018.
2. F. Ouyang, H. Cheng, Y. Lan, Y. Zhang, X. Yin, J. Hu, X. Peng, G. Wang, and S. Chen, "Automatic delivery and recovery system of wireless sensor networks (wsn) nodes based on uav for agricultural applications," Computers and Electronics in Agriculture, vol. 162, pp. 31 – 43, 2019.
3. S. A. Nandhini and S. Radha, "Efficient compressed sensing-based security approach for video surveillance application in wireless multimedia sensor networks," Computers & Electrical Engineering, vol. 60, pp. 175– 192, 2017.
4. E. Intrieri, G. Gigli, T. Gracchi, M. Nocentini, L. Lombardi, F. Mugnai, W. Frodella, G. Bertolini, E. Carnevale, M. Favalli, A. Fornaciai, J. M. Alavedra, L. Mucchi, L. Nannipieri, X. Rodriguez-Lloveras, M. Pizziolo, R. Schina, F. Trippi, and N. Casagli, "Application of an ultra-wide band sensor-free wireless network for ground monitoring," Engineering Geology, vol. 238, pp. 1 – 14, 2018.
5. F. Xia, "Qos challenges and opportunities in wireless sensor/actuator networks," CoRR, vol. abs/0806.0128, 2008.
6. W. A. Aliady and S. A. Al-Ahmadi, "Energy preserving secure measure against wormhole attack in wireless sensor networks," IEEE Access, vol. 7, pp. 84132–84141, 2019.
7. B. Nivedetha and I. Vennila, "Ffbks: Fuzzy fingerprint biometric key based security schema for wireless sensor networks," Computer Communications, vol. 150, pp. 94 – 102, 2020.
8. D. B.D. and F. Al-Turjman, "A hybrid secure routing and monitoring mechanism in iot-based wireless sensor networks," Ad Hoc Networks, vol. 97, p. 102022, 2020.
9. Biswas K., Muthukkumarasamy V., Sithirasenan E., Singh K., "A Simple Lightweight Encryption Scheme for Wireless Sensor Networks," Lecture Notes in Computer Science, vol. 8314, Springer.

AUTHORS PROFILE

Dr. Amarthaluri Thirupathaiah, working as Associate Professor in St. Ann's college of Engineering and Technology, Chirala, India. He has 16 years of teaching experience and published several papers in scopus indexed journals. He received Ph.D (CSE) from Rayalaseema University and M.Tech. (CSE) from Jawaharla Nehru Technological University, India. He is life member of ISTE and CSI. His research interests includes Energy-efficient routing in WSN and Sensor network security.

IB Venkateswarlu, has 11 years teaching experience and published several papers in scopus indexed journals. He received M.Tech. From Jawaharla Nehru Technological University, India. His research interests includes Network security.