International Journal of Applied Electromagnetics and Communication Systems
Volume 2, Issue 02, 2026

Wi-Sense-based Wireless Sensor Network Security Protocol

Sok Hun Kim[1], Gwang Han[2]
Faculty of Mathematics, Kim Il Sung University, Pyongyang, DPR Korea

Research Article

Date of Submission: 25-02-2026 **Date of Acceptance: 23-03-2026** **Date of Publication: 26-04-2026**

ABSTRACT

One of the newest technologies is wireless sensor networks. These networks are responsible for keeping an eye on important environmental factors like temperature, humidity, wind speed, soil moisture content, UV index, sound, etc., and then sending the necessary data to the base station. However, because vital data is being sent, security continues to be the major issue with these networks, particularly when sensor nodes are used in dangerous areas for military purposes. Due to memory and computing power limitations, the majority of sensor nodes in use today lack an effective security standard. Here, we provide a safe and lightweight protocol for wireless sensor applications. A sensor network consisting of WiSense nodes has been used to test the resilient protocol.

KEYWORDS: WiSense, wireless sensor nodes, encryption, decryption, AES, and protocol improvement.

I.INTRODUCTION

Many businesses have created low-power, low-cost wireless sensor nodes that are used to sense environmental factors as a result of advancements in analog and digital technology [1]. Depending on the network design, sensor nodes analyze the data collected and then send only the necessary information to the coordinator or other slave nodes in order to give intelligence for a better knowledge of the environment [2]. Applications include defense monitoring (in reconnaissance situations), automobile emission monitoring, perimeter monitoring, and more. The channel's lack of physical protection and openness provide a security concern.

Because the present encryption standards were created for high-performance devices, they are burdensome for sensors used in sensor networks that have limited processing power and bandwidth. Nevertheless, because of the differences in their architectural design, security methods created for ad hoc networks cannot be easily applied to sensor networks. Sensor networks, on the other hand, have a well defined topology with a central base station to which all the data is sent. They are self-organizing, dynamic, and capable of peer-to-peer data transfer. Using conventional encryption techniques in the sensor would result in the processing and transmission of additional bits, and the network would experience jitter and lag as a result of the significant processing delay [3]. Developing a strong encryption that doesn't need complex cryptographic procedures while yet offering enough security for the intended use is crucial.

Both the central base station and the sensor nodes must implement a reliable but lightweight algorithm. The employment of the AES algorithm for encryption and decryption in sensor networks using the ZigBee protocol was controversial since it requires a lot of processing power and memory, particularly in nodes with limited RAM [4]. Additionally, as the sensor nodes are dispersed geographically and are thus far from direct human contact for extended periods of time, the administration and alteration of the encryption keys in the nodes is a significant factor. To provide a respectable degree of security for low performance nodes, the suggested protocol combines an XOR operation with a key and replacement utilizing an S-Box matrix.

LITERATURE REVIEW

A typical wireless sensor network topology with many sensor nodes linked to the sink node is seen in Figure 1 [5]. A remote console connected to the Internet may operate the sink. Numerous nodes that are dispersed geographically across a wide region make up sensor networks. In order to gather, process, and transmit data to the base station, sensor nodes are made to operate for extended periods of time—possibly months [5].

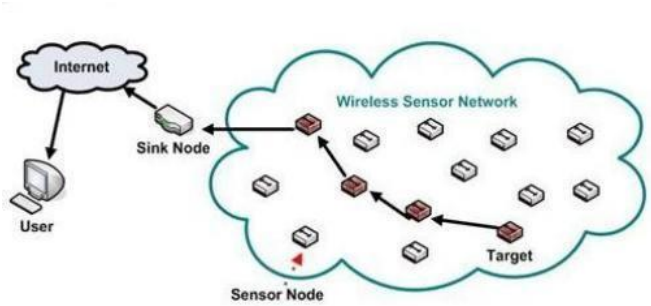


Figure 1: Wireless Sensor Network Architecture.

The nodes of the sensor network may be passive or active. In a passive node, the base station only listens to the node for a predetermined amount of time during which the node sends data back to the base station. Because the node may operate in idle or low-power mode while the base station is not listening, this lowers the node's battery usage. The passive node sensor network has a predetermined duty cycle. Duty cycle is the ratio of the node's low power state to its active state. Better energy efficiency in the sensor network is achieved with a decreased duty cycle, which is desired [6]. The base station continually listens to an active node, which then processes, evaluates, and transmits the collected data back to the base station.

Every node in a mobile sensor network must be able to locate itself using GPS or another relative positioning method. Since sensor nodes may malfunction or be added to the network, network data for ad hoc fixed networks must be constantly updated in real time. The impacts of shallow fading, the number of connections, the communication length, the mode of propagation, and the minimum acceptable quality of service must all be taken into consideration while designing the software and transmission protocols. Redundancy in the number of nodes is necessary to prevent the network from going down since the sensor nodes are very sensitive to external factors (such as wind speed, humidity, rain, etc.).

Since most nodes operate in hostile environments and it is crucial that their existence stay undiscovered, security mechanisms must be included into the nodes' original architecture rather than being added later. They have a finite amount of energy (between 20 and 30 joules). In order for the information to be sent to the central node, the incoming data must be merged with the local information due to the many sources of data. The network must have very little transmission delay in order to distribute the massive volumes of data that are gathered and analyzed. It is necessary to create security procedures that guard the network from infiltration and spoofing.

2.1. WiSense Node Architecture



Figure 2: WiSense Node –courtesy WiSense [7].

The WiSense WSN1101L [7] wireless mesh node is seen in Figure 2. It is made up of the radio module (CC1101) and the microcontroller module (MSP430G2955). An antenna is used by the radio module to receive and send data. TI's ultra-low power

microcontroller, the MSP430G2955, has 4 KB of static RAM and 56 KB of flash memory. One microampere is the lowest possible standby current. The microcontroller's operating voltage ranges from 1.8 V to 3.6 V. 10-bit ADC channels are included within the module. The module has SPI, I2C, and UAR peripheral capabilities. The radio module has an SPI and GPIO interface, while the sensors have a UART, I2C, SPI, or GPIO interface. The clock is provided by an internal 32 KHz crystal.

The CC11011 radio module works at frequencies lower than 1 GHz. It is an inexpensive transceiver that may be used in remote sensing applications. It operates in the 300-348 MHz, 387-464 MHz, and 779-928 MHz bands [8]. Flexible (amplitude-shift keying) and on-off keying (OOK) The two modulation methods that are supported for RF transmission and reception are ASK shaping. The range of data rates is 0.6 to 600 kbps. The onboard antenna is a 117 mm long omnidirectional whip antenna. The circuit's clock is provided by an on-board, very accurate 26 MHz crystal.

2.2. Issues in Implementing AES in Sensor Nodes

Symmetric approaches lack nonrepudiation and have problems with key agreement and key scalability [9]. AES is said to be sluggish and needs a large amount of memory to store the lookup tables. AES offers higher mean energy-latency product values; for 128-bit AES encryption, the energy per bit for the encryption operation is about 151 nJ/bit. [10]. According to Hodjat, the whole program including data retrieval, encryption, and computation requires a full 704 clocks, even though the AES encryption only requires 11 cycles. It was discovered that the decryption procedure used 20–30% more energy. Additionally, symmetric methods like AES rely heavily on architecture [11].

2.3. Limitations in implementing AES in Wi Sense Nodes

The 16-bit RISC architecture-based TI MSP430G2955 microcontroller serves as the foundation for the WiSense [7] node under discussion [12]. Implementing symmetric encryption techniques on 16-bit microcontrollers such as the MSP430 has significant limitations. The node in question is a 16-bit microcontroller, however the majority of these algorithms are designed for 32-bit CPUs. The TI MSP430's restricted instruction set prevents the Rijndael cipher block from being tailored for 32-bit CPUs. Additionally, even if these improvements are implemented, the bulk of the node's RAM will be used. The instruction set of the MSP430 is straightforward yet constrained. While accessing the physical memory is expensive and may take up to three cycles, most register operations only need one cycle. For instance, using the TelosB note's conventional 128-bit AES implementation [13] would need 1.994 ms for encryption and a further 2.365 ms for decryption [14].

II. NETWORK ARCHITECTURE

3.1. Wi-Sense Network

A multihop mesh WiSense Sensor-Actuator Network (WISAN) with 25 nodes and 72 sensors running at 2405Mhz with a transmission power of 5 dbm has been considered. The sensor nodes gather information on pressure, voltage, and temperature. The MAC ID of the coordinator node is 0xDEADBEEFFEEDDADD [15]. The sensor data for the last and final update time tags is shown in Table 1. The nodes' addresses and locations as they are geographically dispersed across the environment are shown in Table 2. The TI MSP430G2955 microcontroller serves as the foundation for the node. There are two PCBs in the module. The microcontroller is located on one PCB, while the CC1101 radio is located on the other. The microcontroller PCB has 4 KB of SRAM, 56 KB of flash, and a standby current of under 1 micro amp (in LMP3). In order to offer both power and serial communication, it contains an optional serial to USB interface on board. With a 32 kHz crystal serving as a real-time clock, it features peripheral support for SPI, I2C, and UART [7].

Table1: Sensor Information

Name	Id	Last Update	Next Update
0002	1	2015-10-18 04:04:57	-
0013	1	2015-10-18 02:03:13	2015-10-18 04:03:13
0019	4	2015-10-18 01:38:33	2015-10-18 04:38:33
0014	1	2015-10-18 01:16:00	2015-10-18 04:16:00
0005	3	2015-10-17 23:49:14	2015-10-18 00:49:14

0004	2	2015-10-17 22:38:40	2015-10-18 00:38:40
000F	2	2015-10-17 18:03:57	2015-10-17 18:08:57
0013	5	2015-10-17 17:03:04	2015-10-17 20:03:04
0006	2	2015-10-17 16:32:14	2015-10-17 17:32:14
0010	3	2015-10-17 15:48:41	2015-10-17 15:49:11
0010	2	2015-10-17 15:48:40	2015-10-17 15:50:40
0010	1	2015-10-17 15:48:40	2015-10-17 17:48:40
0004	3	2015-10-17 15:48:10	2015-10-17 16:48:10
0008	3	2015-10-17 15:48:04	2015-10-17 17:48:04
0018	2	2015-10-17 15:11:41	2015-10-17 18:11:41
0002	2	2015-10-17 12:50:35	-
0004	1	2015-10-17 12:43:50	-
0008	1	2015-10-17 09:25:46	-
000D	4	2015-10-17 09:04:03	2015-10-17 12:04:03
0017	4	2015-10-17 08:59:58	2015-10-17 09:59:58
0017	3	2015-10-17 08:59:58	2015-10-17 09:04:58
0017	1	2015-10-17 08:59:57	2015-10-17 10:59:57
0008	2	2015-10-17 08:53:28	-
0011	3	2015-10-17 07:30:49	2015-10-17 10:30:49
000D	2	2015-10-17 07:30:45	-
0006	4	2015-10-17 06:42:23	-
0012	1	2015-10-17 05:48:30	2015-10-17 05:49:00
0011	4	2015-10-17 04:38:00	2015-10-17 04:39:00
006	3	2015-10-17 03:07:38	-
000C	3	2015-10-17 02:02:56	-
0009	1	2015-10-17 01:10:27	-
000F	6	2015-10-17 00:32:06	-
0005	2	2015-10-16 23:47:16	2015-10-16 23:48:16
0014	4	2015-10-16 23:46:03	2015-10-16 23:48:03
0019	3	2015-10-16 23:34:20	2015-10-16 23:34:50
0019	1	2015-10-16 23:34:19	2015-10-16 23:34:49
000F	5	2015-10-16 23:30:06	2015-10-16 23:32:06
000F	1	2015-10-16 23:30:04	2015-10-16 23:32:04
0014	3	2015-10-16 21:05:30	2015-10-16 21:10:30
0007	1	2015-10-16 18:47:45	2015-10-16 19:47:45
0007	4	2015-10-16 18:47:45	2015-10-16 19:47:45
0014	5	2015-10-16 17:26:13	2015-10-16 17:26:43
0016	1	2015-10-16 17:21:28	2015-10-16 17:23:28
000C	4	2015-10-16 17:12:18	2015-10-16 17:12:48
000C	2	2015-10-16 17:12:17	2015-10-16 17:17:17
0007	3	2015-10-16 16:17:19	-
0011	2	2015-10-16 15:40:15	2015-10-16 15:42:15
0018	4	2015-10-16 14:21:24	-
0007	2	2015-10-16 12:36:25	-
000D	1	2015-10-16 12:33:41	2015-10-16 13:33:41
000E	3	2015-10-16 11:40:07	2015-10-16 13:40:07
000D	3	2015-10-16 09:55:55	-
000C	1	2015-10-16 09:28:08	-
0003	1	2015-10-16 09:19:50	2015-10-16 10:19:50
0006	1	2015-10-16 08:34:04	-
0017	2	2015-10-16 07:57:38	-
000F	4	2015-10-16 07:56:27	-
0018	3	2015-10-16 07:52:37	2015-10-16 07:54:37
0018	1	2015-10-16 07:52:36	2015-10-16 07:53:36
000E	2	2015-10-16 06:23:18	2015-10-16 06:28:18
0019	2	2015-10-16 05:01:43	-
0013	3	2015-10-16 04:21:11	-
0013	2	2015-10-16 03:28:17	-
0013	4	2015-10-16 03:10:35	2015-10-16 03:11:35
0014	2	2015-10-16 03:04:01	2015-10-16 03:09:01
000F	3	2015-10-16 01:42:43	-

0011	1	2015-10-16 01:20:02	-
000E	4	2015-10-15 23:58:14	2015-10-16 00:03:14
000E	1	2015-10-15 22:50:11	-
0011	5	2015-10-15 19:15:10	2015-10-15 19:16:10
0005	1	2015-10-15 17:46:07	2015-10-15 18:46:07
0005	4	2015-10-15 13:00:01	2015-10-15 5:00:01

-courtesy WiSense [21]

Table 2: Node Information

Name	*Location*	*MAC Address*	*Short Address*
0002	Blk-B	0x0002000200020002	0x0002
0003	North Wing	0x0003000300030003	0x0003
0004	Corridor-B	0x0004000400040004	0x0004
0005	B-A	0x0005000500050005	0x0005
0006	South Wing	0x0006000600060006	0x0006
0007	Blk-A	0x0007000700070007	0x0007
0008	Car Park	0x0008000800080008	0x0008
0009	Nw Lab	0x0009000900090009	0x0009
000A	Corridor-A	0x000A000A000A000A	0x000A
000B	South Wing	0x000B000B000B000B	0x000B
000C	South Wing	0x000C000C000C000C	0x000C
000D	Blk-B	0x000D000D000D000D	0x000D
000E	South Wing	0x000E000E000E000E	0x000E
000F	Blk-B	0x000F000F000F000F	0x000F
0010	North Wing	0x0010001000100010	0x0010
0011	North Wing	0x0011001100110011	0x0011
0012	Corridor-B	0x0012001200120012	0x0012
0013	Blk-A	0x0013001300130013	0x0013
0014	Nw Lab	0x0014001400140014	0x0014
0015	Nw Lab	0x0015001500150015	0x0015
0016	Corridor-A	0x0016001600160016	0x0016
0017	ECE Dept	0x0017001700170017	0x0017
0018	Nw Lab	0x0018001800180018	0x0018
0019	Corridor-B	0x0019001900190019	0x0019

3.2. Firmware Elements

The Reduced Function Device (RFD), Full Function Device (FFD), Coordinator, and Gateway are the firmware components [16]. In essence, a Reduced Function Device (RFD) is a mote with sensors and firmware code for both local and distant processing. The RFD transmits the available sensor data to the FFD. Similar to an RFD, a Full Function Device (FFD) serves as an intermediary relay node. A adjacent FFD receives information from an RFD and handles the routing of that information to its surrounding FFDs. Until the coordinator is reached, this chain continues. The system's master device and control point is the PAN coordinator. There may be several systems in the same location, but each system will have its own coordinator. When RFDs and FFDs join that particular system, they collaborate with the coordinator. Each hub is assigned a PAN address by the coordinator that is

valid inside the system. The gateway receives messages from both full-function and reduced-function devices. The

coordinator prepares control messages, although sensor data is often transmitted to the gateway for LAN/WAN transmission. The gateway receives explicit control messages, which are then sent to the nodes via the coordinator.

IMPLEMENTATION

This study suggests effective communication between CR in the Advanced Encryption Standard (AES) using Cipher Block Chains, where the first block's encryption result is utilized to encrypt the subsequent block. The procedure is carried out repeatedly. Such a design need feedback. SubBytes, Shift Rows, Mix Columns, and Add RoundKey are the procedures used in each encryption round [17]. Ten rounds overall, with the last round omitting the Mix Columns phase. Each step is mathematically inverted at the recipient's end, but the order is switched. We take into account a block size of 128 bits. The other three phases are all linear, with the exception of the SubBytes step. The Galois inverse of an 8-bit value that is included in the Galois Field GF (28) is computed in the sole nonlinear step (SubBytes). The look-up table approach has one significant drawback while being faster: it takes up 256 bytes of storage, without accounting for the RAM required to address and store the results. [18]. The substation must handle each of the 16 bytes in the block independently and concurrently. In real implementations, however, this would result in 16 copies of the S-Box table for a single round and 160 copies for the whole encryption process in a pipeline implementation [17]. The SubBytes step alone would need 40960 bytes of memory, which is costly for sensor nodes.

4.1. Proposed Protocol

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 3: AES S-Box [19].

An AES S-Box is seen in Figure 3 [19]. A 16*16 S-Box [20] will be incorporated into each sensor node in the network, taking up a total of 2 kilobits of flash memory. Now consider two nodes that want to exchange data, such as humidity, temperature, wind speed, soil moisture content, UV index, or sound levels. The value to be sent is first substituted with the corresponding element in the 16*16 S-Box using the substitution method. The timestamp's hour, minute, or second value, the node name, or the last two bits of its MAC address may all be used as keys to EX-OR the replacement value. The received data is once again EX-ORed with the key at the receiving end before being mapped using the S-Box to determine the unencrypted value. To save memory, we merely take into account two digits of the node name, timestamp, or MAC address as the key. An additional flag indicating whether the hours, minutes, or seconds are regarded as the time stamp value should be sent if we were to utilize the timestamp as the key. For a lightweight sensor network like Wi-Sense, where it is difficult to include more sophisticated security algorithms, the aforementioned suggested security protocol uses the least amount of memory—just over two kilobits—along with some memory required for the runtime code. The robustness of the protocol is confirmed, and it creates an improved security protocol for wireless sensor networks.

4.2. Example

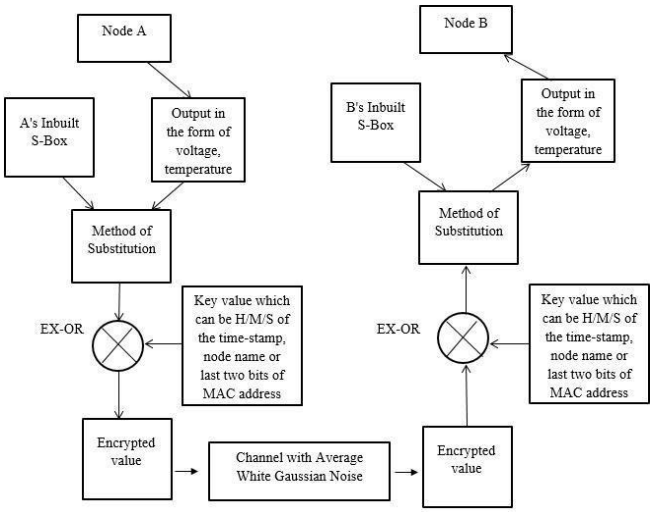


Figure 4: High level Block Diagram of the prosposed encryption and decryption process.

The block diagram in Figure 4 illustrates the high-level implementation of the suggested encryption and decryption procedure. The time-stamp, node name, or MAC address may all be used as the Ex-OR operation's key. On the other hand, we see the MAC address as the key. If we were to use the time-stamp as the key, we would need an extra 2-bit flag to indicate whether the time-stamp value is in hours, minutes, or seconds. This is vulnerable to channel noise in addition to adding an additional burden. Bit flipping and the incorrect time stamp being utilized at the receiver for the Ex-OR operation in the decryption process might result from any interference in the channel.
We assume that node 08, which has a MAC address of 0x0008000800080008 (from Table 2), transmits the 8-bit data 0xD5 along with the associated encryption and decryption procedure. After the Sub Bytes operation, the replaced value is 03 in hexadecimal notation since the data first travels through the S-Box substitution procedure. The last two digits of that node's MAC address (0x08) are then EX-ORed with the replaced value. The final encrypted value that has to be sent is 0x0B, or eight bits. The information is packetized. The physical address of the source node, which is a mix of the sender node's MAC address and IP address, is also included in the packet. The 8-bit encrypted data is EX-ORed with the last MAC address digits at the receiving node after deframing. The outcome in this instance is 0x03. The decrypted value of 0xD5 is now obtained by mapping this result to the inverse S-Box table.

CONCLUSION

It is therefore essential to provide a simple but reliable security protocol for hardware onstraint sensor nodes due to the increasing use of wireless sensor networks for environmental parameter monitoring and measurement. Performance and security of the created protocol were determined to be adequate when tested on WiSense nodes based on the MSP430. Sub Bytes and EX-OR together provide a robust defense against unwanted source infiltration. To provide wireless sensor platforms the best possible secure communication infrastructure, further improvements may be made.

REFERENCES

1. 1. WIRELESS SENSOR NETWORKS: A SURVEY, COMPUTER NETWORKS 2002; 38: 393-422, IF AKYILDIZ, W SU, ET AL.
2. 2. WIRELESS SENSOR NETWORKS, T. RAGHAVENDRA, 2004.
3. 3.H AL-SAKIB KHAN PATHAN, WIRELESS SENSOR NETWORK SECURITY: PROBLEMS AND DIFFICULTIES, 2006.
4. 4.SB SHAMMI DIDLA, WIRELESS SENSOR NETWORKS AND EMBEDDED DEVICE AES OPTIMIZATION.
5. 5. VS RAJKUMAR, ISSUES AND APPLICATIONS OF WIRELESS SENSOR NETWORKS, INTERNATIONAL JOURNAL OF COMPUTER TECHNOLOGY AND APPLICATIONS, 2012, 3: 1667-1673.
6. 6.P ARCHANA BHARATHIDASAN, AN OVERVIEW OF SENSOR NETWORKS.
7. 7.WISENSE, WSN1101L DATASHEET, 2015.
8. 8. TEXAS INSTRUMENTS, 2015; TI.COM.
9. 9. TOIVONEN, SYMMETRIC AND ASYMMETRIC CRYPTOGRAPHY IN FALSE REPORTS, NETWORK SECURITY SEMINAR, 2007.
10. 10. POTLAPALLY, EXAMINING SECURITY PROTOCOLS' ENERGY USE, INTERNATIONAL SYMPOSIUM ON LOW POWER ELECTRONICS AND DESIGN, 2003, 25–27.
11. 11.Y WANG, IEEE COMMUNICATIONS SURVEYS & TUTORIAL, A SURVEY OF SECURITY ISSUES IN WIRELESS SENSOR NETWORKS, 2006; 8.
12. 12. TEXAS INSTRUMENTS (TI), 2013.
13. 13. CROSSBOW, "TELOSB DATASHEET,"
14. 14. M. T. J. P. K. M. D. S. PIERANGELA SAMARATI, "INFORMATION SECURITY THEORY AND PRACTICES: SECURITY AND PRIVACY OF PERVASIVE SYSTEMS AND SMART DEVICES," 4TH IFIP WG 11.2 INTERNATIONAL WORKSHOP, WISTP, PASSAU, GERMANY, 2010.
15. 15. [ONLINE] WISENSE, HTTP://WWW.WISENSE.IN/WSNGUI/, 2013.
16. 16. WISENSE, WISENSE DOCUMENTATION, WISENSE, 2014.
17. 17.V DAI YAMAMOTO, AES S-BOX-BASED GLITCH PUFS ON FPGAS: PERFORMANCE AND SECURITY ASSESSMENT, INTERNATIONAL CONFERENCE ON SECURITY, PRIVACY AND APPLIED CRYPTOGRAPHY ENGINEERING, 2012.
18. 18.E. TRICHINA, AES SUBBYTE TRANSFORMATION USING COMBINATIONAL LOGIC DESIGN ON MASKED DATA, INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH, 2003.
19. 19.W. STALLINGS, PRINCIPLES AND PRACTICES OF CRYPTOGRAPHY AND NETWORK SECURITY, PRENTICE HALL, 2011.
20. 20. SELENT, ADVANCED ENCRYPTION STANDARD, INSIGHT: RIVIER ACADEMIC JOURNAL, 2010; 6. 21. WISENSE, WISENSE, 2013 [ONLINE]. [ONLINE] 22.WISENSE, WISENSE, 2013. HTTP://WWW.WISENSE.IN/WSNGUI/NODES. HTTP://WWW.WISENSE.IN/WSNGUI/SENSORS IS ACCESSIBLE.