ISSN:0975-9646

Methodology for Deploying a Security Framework in Mission Critical Infrastructure Based Wireless Sensor Networks

Sudipto Roy [#1], Manisha J Nene [#2]
[#] *Department of Computer Science,*
Defence Institute of Advance Technology (Deemed University)
Pune-411025, India

Abstract-This paper presents a methodology on design of a security framework for Wireless Sensor Network (WSN). This framework is proposed for an infrastructure based deployment of the sensor nodes for mission critical application like static military deployment. Major security prerequisite, like confidentiality, integrity, freshness and authentication are being addressed. In addition a key management and distribution methodology is also been proposed in the solution. The paper also outlines the factors that have to be considered while designing a framework. It also gives an insight in the available platform that are available for the design of the framework and considerations that have to be kept in mind while setting up an experimental setup for testing the security framework. The major focus of this proposal is to achieve the desired level of security within the resource restriction imposed by WSN.

Keywords: *Symmetric encryption, Block cipher, Message authentication, Replay Attack, Key Management Mechanism, TinyOS, Contiki OS, Cooja Simulator.*

1. INTRODUCTION

With the improvement in the configuration of the wireless sensor nodes, in term of increased computational power, memory available for programming, etc., the versatility of wireless sensor nodes have increased many folds. They now find utilisation in everyday life starting from mundane use like management of appliances at home to security sensitive operations like use in the military or health care management [1].

With the kind of information being handled by WSN gaining importance, security becomes a matter of prime concern. WSN have a lot of similarities with traditional computer networks. However, the resource restrictions in terms of available memory, computational power and limited power availability, makes traditional security measures too costly to be implemented for WSN.

With every passing day new kind of attacks are being envisaged and experienced in a WSN. Yet the primary security requirement for any kind of network is that of data confidentiality, authentication, data integrity, authorisation and freshness. In short, the security services in a WSN should make an endeavour to protect the communicated information over the network, the resources from attack and unauthorised disclosure [2][3].

This paper proposes a security framework which tries to fulfil the major security concerns. We have categorised the security framework under some major modules, namely, cryptographic algorithm, modes of operations, authentication mechanism and key management. Related studies have been carried out for the respective modules and the most efficient mechanism/ protocol/ algorithm, in context of WSN resource limitation has been selected.

Deployment pattern plays a very crucial role in WSN. The deployment pattern is governed by the user requirement. We propose this security framework for static military requirements like border fencing monitoring or mine field surveillance.

We have also surveyed the existing security framework for WSN and brought out how our proposal is different from the other security framework.

The rest of paper is organized as follows. Section 2 provides summary of related work or our concern for wireless sensor network. Section 3 presents the proposal of the security framework. Section 4 provides a brief methodology for the deployment of the security framework on a simulated environment using the desired platform OS. Section 5 the paper has been concluded providing the future research directions.

2. RELATED STUDIES

The advantages gained from the security framework cannot be overlooked, but the same comes at the cost of increased latency, reduced available memory and introduces computational overhead. Any security framework can be broken down in to different modules. Each module can be entrusted with different security objectives.

Cryptographic technique is the most important module of the security framework. Any cryptographic algorithm must be used in conjunction with the appropriate mode of operation, which ensures the encryption of long messages. It also ensures that an encrypted plain text will result in different ciphertext on multiple encryptions with the same key. Message authentication code (MAC) ensures the integrity and authentication of the data. Management of Key is also an important aspect of any framework.

2.1 Cryptographic Algorithm

In this subsection, cryptographic algorithm has been covered. Any cryptographic algorithm is rated against the following performance factors [4], namely, tenability, computational speed, key length, encryption ratio, security issue, time and throughput of data against attacks.

In addition to the above, in case of WSN, the performance parameters [5] that decide the efficiency of a security framework are code size, power consumption and memory requirement.

Cryptography is primarily classified under symmetric key or private key and asymmetric key or public key [4]. In symmetric cryptography a single key is used for encryption. In asymmetric encryption we use two set of keys. One set of keys known as public key is used for verifying digital signature and to encrypting the plaintext. The private key on the other hand is used to make digital signatures and to decrypt the ciphertext [6]. Symmetric key encryption is based on transposition and substitution cipher block and asymmetric key encryption is based on mathematical problems of discrete logarithm, elliptic curve relationships and integer factorization [7].

Majority of researcher community opines that in resource constraint platform like WSN, symmetric key encryption is favored. Though a lot of research has been done in terms of using asymmetric key algorithms in WSNs in terms of enhancing the energy efficiency [8][9][10], by the use of efficient energy techniques like elliptic curve cryptography, yet the results are not encouraging. Asymmetric key encryption is resource intensive in terms of consumed power and computational latency and is thus not a preferred choice over symmetric key encryption.

In case of symmetric key encryption, the encryption and decryption process uses the same key. Thus key management becomes a matter of prime importance [11].

Further studies are based on symmetric key encryption only. Symmetric key encryption is further divided into two sub classes namely block cipher and stream cipher [7]. In block cipher we take a block of n bits and convert them to n block of cipher text, where n is called the block length. Some examples of block cipher are AES [12], LED Block Cipher [13], Skipjack [14] [15], Triple DES [16], Blowfish [17] etc.

The major difference between block cipher and stream cipher is that block cipher treats data in multiple of a fixed size block where as in stream cipher the data is encrypted on the fly. Some examples of stream cipher are SOSEMANUK [18].HC-128 [19], Trivium [20], Rabbit [21], Salsa20/12 [22], MICKEY 2.0 [23], etc.

Gustavo et al.,[24] has brought out the various attack that are possible on the stream cipher, namely Correlation Attack, Algebraic Attack, Slide Attack, Exhaustive Search Attack, Distinguishing Attack, Fault Attack etc. AUDIA S. et al., [25] had stated that stream cipher is difficult to be implemented in software and suffers from synchronization mismatch. The effect of synchronization mismatch has an avalanche effect on the data stream thereafter. Due to the numerous problems with stream cipher in WSN, we only concentrate for the block cipher for further analysis.

Any cipher selected should be in conjunction with the available resources. The restricted code and program memory, limited computational capability and power requirement must be adhered to. Mickael Cazorla et al., [26] have benchmarked eighteen different block cipher. Out of the eighteen that have been surveyed and benchmarked, thirteen are light weight block cipher protocols, i.e. it uses 32, 48 or 64 bits as the block. Other five protocols which were benchmarked were conventional block cipher and uses 64/128 bits as the block size. The block cipher being considered by Mickael Cazorla et al., [26] are AES-128 [12], CLEFIA- 128 [27], DESXL[28], HIGHT[29], IDEA [30], KATAN [31], KLEIN[32], LBLOCK[33], LED[34], mCrypton [26], MIBS [35], Noekeon [36], Piccolo [37], PRESENT [38], TEA & XTEA [39], TWINE [40], SEA [41] and SKIPJACK [15].

Mickael Cazorla et al., [26] from his experimental setup benchmark all the different block cipher on the parameters of the computational overhead, RAM and ROM utilisation. He concludes that out of the eighteen protocols under consideration, Piccolo, XTEA, AES or TWINE shows good performance considering code size and cycle count trade off.

The next logical conclusion will be to consider the security vulnerabilities or the cryptanalysis possible on the above shortlisted block cipher methods.

Kitae et al., [42] state that when Piccolo-80 and Piccolo-128 are subjected to biclique cryptanalysis on full round versions then the probability of exposing the key, on fewer error and trial increases.

XTEA has been highly analysed for security vulnerability. Many papers have analysed the security of XTEA. The most recent publications of Jiazhe et al., [43], present an impossible deferential attack on 23rd round of XTEA. Yu Sasaki et al., [44], demonstrated a meet-in-the-middle attack, with nine known plaintexts which can be applied against twenty five rounds of XTEA.

Mustafa et al., [45] also demonstrate a biclique cryptanalysis on TWINE-80 and TWINE-128.

AES-128 is also shown some vulnerability to cryptanalysis. Andrey et al., [46], has demonstrated that AES can be subjected to biclique cryptanalysis, but the time complexity of such analysis is too high to be of practical importance. Patrick Derbez et al., [47] have demonstrated a meet in the middle attack on seven rounds of AES within practical time and memory restrictions. However, AES in its native form employs ten rounds, so practicality of such a cryptanalysis is negated out.

AES-128 proves its strength over all the block cipher considered. However, we need to critically examine AES on power efficiency. Law et al., [48], brings in the benchmark comparison of AES with other block cipher, namely ciphers is RC5 [49], RC6 [50], KASUMI [51], Camellia [52] and MISTY1 [53]. All this block ciphers have proved their immunity towards differential and linear cryptanalysis and thus are at par with AES-128 as per the security standards are considered.

Law et al., [48], benchmarking clearly shows that MISTY1 is superior to almost all the discussed block cipher methods as per speed and size optimisation is concerned. Speed optimisation leads to less no of clock cycle and thus less power consumed. Thus we conclude that MISTY1 is the best choice of block cipher available under the resource limitation of WSN.

2.2 Operation Modes

When working with block cipher the modes of operations are an important consideration. Block cipher are basically designed to handle several block of data. There is a risk of producing the same cipher text using the same plain text and the key [8]. To prevent this phenomenon, block cipher introduces the mode of operations [54].

Modes of operations can be used for many different purposes like mode for encryption, mode for data integrity, modes that achieve both encryption and integrity, modes that gracefully recover from errors in transmission, etc. [3]

In this subsection we shall only cover the mode for encryption. According to NIST, there are five different modes of operations to provide encryption, namely, the Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). Our study in this field, tries to find out the best suited mode of operation for encryption, for block cipher.

Phillip et al., [55], has compared all the mode of operations under semantic security (SEM CPA). Semantic security was introduced by Goldwasser et al [56]. It emphasis is on the fact that security settings is not absolute as proposed by Shannon [57]. Rather it is dependent on the computational effort made by an adversary.

ECB was designed for use with symmetric block cipher, DES. However ECB lacks any secrecy quality and produces the same cipher text for the given plaintext under the same key. We conclude that ECB is recommended only for single block messages [3] and will not achieve any desirable privacy.

CBC, CFB and OFB are generally clubbed together as they share some common characteristics like the use of an Initialisation Vector (IV). In the SemCPA sense, all the above modes are secure, if the user uses a random IV. Alternately, in the SemCPA sense, none of the modes are secure, if one merely uses a nonce IV [55]. In classical sense it is very difficult to achieve a random IV thus adding to the security vulnerability of CBC, CFB and OFB.

CTR mode is treated separately as it provides a nonce based notion for security, it has a different historical context for IV as it is not an n-bit string. CTR can be considered as the preferred mode for confidentiality as it provides a guaranteed provable-security [55].

Jongdeog Lee et al., [58] give a different perspective on comparison of the operations modes. The operation modes are compared on the basis of the memory requirement, energy efficiency and efficiency in gaining synchronisation. OFB has the highest energy efficiency, highest memory efficiency and desirable fault-tolerance characteristics, i.e. a corresponding plain text is only affected by the corresponding ciphertext error. However OFB, to regain synchronisation in case of any loss, is dependent on an external mechanism. This same principle also applies for CTR, but CTR regains synchronisation faster than OFB, as CTR operations are parallelisable. CTR is only next to OFB in terms of energy-efficiency. However, CTR has the highest RAM usage among all the modes, still one can justify the high RAM usage against the potentially great savings in resynchronisation.

We conclude that the use of CTR mode is the best choice for secrecy mode of operation for block cipher.

2.3 Message Authentication Mechanism

Authentication mechanism deals with two different tasks, namely, message authentication and sender authentication. Here our related study focuses on message authentication, with the aim to figure out the best available algorithm for message authentication code (MAC).

The MAC techniques can be broadly classified under three different heads namely conventional MACs, Nonce based MAC and Nonce based Authenticated Encryption with Associated Data (AEAD) [55]. Under conventional MAC, we have Cipher Block Chaining Message Authentication Code (CBC-MAC), Cipher-based Message Authentication Code (CMAC) and Hash Based Message Authentication Code (HMAC). Under nonce based MAC we have Galois Message Authentication Code (GMAC) whereas Counter Mode Cipher Block Chaining Message Authentication Code (CCM) and Galois/Counter Mode (GCM) fall under the nonce based AEAD scheme.

Our study focuses on understanding all the different mechanisms in context of WSN. We try to figure out the most efficient one in terms of security, storage space and computational efficiency.

Karl Brincat et al., [59] demonstrate some serious security flaws in CBCMAC like birthday attack, key guessing attacks and cut and paste attack. It provides a very restricted domain of operation in which the input to CBCMAC must be a positive multiple of the block size. Practically this may lead to padding extra bits, thus introducing computational overhead and deteriorate efficiency. Antoine Joux et al., [60] demonstrated forgery attack and key recovery attack on CBCMAC.

CMAC has evolved from CBCMAC and proves better provable security. However, the biggest security issue with CMAC is the frequent change of keys. CMAC also suffers in efficiency as it is predominantly serial in nature and its performance is limited to the underlying block cipher. Mitchell et al., [61] has demonstrated birthday attack on CMAC.

HMAC [62] provides a MAC by a simple keying function to a cryptographic hash function. The strength of the HMAC depends upon the underlying compression function being truly random [63]. HMAC uses an iterative structure and are subject to birthday attack [64]. However such attack depends on the underlying hash function. The hashing function used in case of HMAC is MD4, MD5 or SHA1. Majority of the cryptanalysis work on HMAC concentrates on the properties of the hashing function being truly a random function [65]. HMAC-MD4 is not a true pseudo random function and is subjected to direct attack. HMAC-MD5 and HMAC-SHA1 have no vulnerability enlisted. But increasing the complexity of the underlying hash function reduces the efficiency in computation.

GMAC and GCM [66] are similar in principle. GCM in addition to GMAC also provides encryption. We limit our study to GCM only. Galois/Counter Mode (GCM) combines counter mode encryption and Carter Wegman message authentication [67] to provide AEAD. Niels et al., [68] shows two major weaknesses in GCM due to the use of small size authentication tag. The first weakness significantly raises the probabilities of successful forgery. The second weakness gives away the authentication key if one manages to create successful forgeries.

CCM is a nonce based AEAD mode of operation. It uses the counter mode for encryption and the AES based CBC-MAC for authentication. Pierre-Alain Fouque et al.,[69] had found some vulnerability in CCM due to the use of the repeatable nonce, but the same was rectified in the later versions. However, CCM has got a time penalty and is slow in execution. Further the complexity of implementation poses serious performance deterrence. In spite of the above, CCM finds wide spread application and has been accepted as a standard in IEEE 802.11, 802.15.4, IKEv2 etc.

2.4 Key Management

Researchers are of the opinion that it is very difficult to find a key management mechanism that is optimal for all kind of topology of WSN. Thus it is up to the application developer to find the best key management mechanism, which is specific to the deployment pattern and the application being designed [70]. Our primary focus while conducting a literature survey was to focus on finding the best suited key management mechanism for our deployment.

Suman et al., [71] gives a detailed classification of the different key management schemes that are available and protocols that have been designed under those designs.

Chih-Chun Chang et al., [72] has brought out the differences in all the key management schemes and compared them on the basis of their storage performance, node capture resilience, key connectivity and scalability.

JOHNSON C. LEE et al., [73], suggests that one need to keep in mind the trade-offs when deploying the required key management scheme. Schemes that provide a high degree of resilience and scalability often increases the complexity of the code and increases implementation complexity.

Key management is a widely researched topic. It is very difficult to compare all the schemes and determine which one is better as all protocols have different trade-off. We have to focus on the key requirement and then home on to the best available scheme.

2.5 Security frameworks

Gaurav Sharma et al [74], proposes that most of the framework technique are classified based on the cryptographic technique being used with an associated key distribution mechanism. The different kind of cryptographic technique used and the associated security framework for WSN has been summarised in Fig. 1. We provide a brief on all the techniques with the associated security vulnerability.

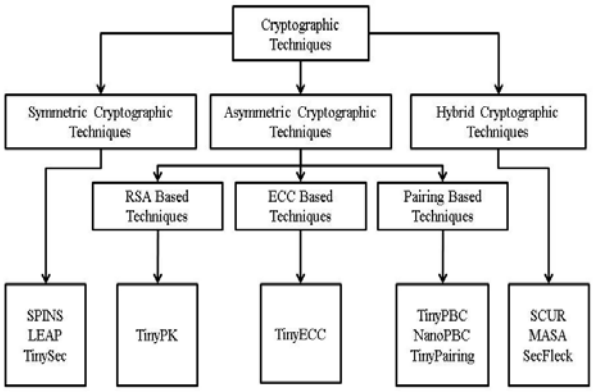

Fig. 1. Different Cryptographic technique [76]

SPINS, TinySec and Localized Encryption and Authentication Protocol (LEAP) are based on symmetric key encryption.

SPINS – Perrig et al., 2001 suggested a security framework optimised for limited resource environment, like WSN, using secure building block like SNEP and μTESLA. This scheme provides good resilience against node capture and is able to revoke keys. However, the suggested framework is not scalable. It also makes the base station the main target for attack. It also doesn't provide security against replay attack and Denial of Service (DoS).

Localized Encryption and Authentication Protocol (LEAP) - Zhu et al. [75] proposed a security protocol known as LEAP. This protocol provides network data processing in form of data aggregation and passive participation. This framework also supports different security mechanism for different type of communication like unicast, local broadcast and global broadcast. However this protocol suffers from some short comings. Several assumptions like static sensor node are not realistic. The protocol also assumes that hundreds of bytes are available at each node for storing the keying materials.

TinySec - Karlof et al., [76] designed a generic and lightweight link-layer security protocol which can be easily integrated in to any sensor network applications. The same is also integrated in the official version of TinyOS. It has the property of using low bandwidth and energy and has low latency thus being able to extend functionalities into higher level protocol. However, careful analysis of TinySec reveals some shortcomings. This protocol doesn't address the issue of replay attack. TinySec uses SKIPJACK block cipher for CBC/CBC-MAC. Lars Knudsen et al. 1999 demonstrated a truncated differential attack against 28 rounds of Skipjack cipher.

In the Asymmetric cryptographic frameworks, we have three basic type of security framework technique namely RSA based, Pairing based and ECC based.

TinyPK [77] is an implementation based on RSA. This deals with both authentication and key agreement between two entities in WSN. However RSA is computationally

intensive and Carmen et al., 2009 has demonstrated that such calculation requires thousand joules for simple multiplication operation function even for 128 bit implementation.

TinyECC [78] is an example of ECC based cryptography framework. ECC techniques are assumed to be better than RSA techniques due to smaller key size for same level of security thus requiring less storage and less bandwidth. This subsequently reduces the power consumption. TinyECC provides a design and implementation methodology for sensor nodes to implement ECC based public key cryptography. Though the energy consumed by ECC based techniques is less than RSA based technique, yet they consume significant amount of energy which is many folds more than symmetric cryptography framework [79].

Asymmetric cryptographic frameworks using pairings is a related field to ECC. Use of this scheme makes the cryptographic framework more efficient and robust. TinyPBC (Oliveira et al., 2008), NanoPBC (Aranha et al., 2009) and TinyPairing (Xiong et al., 2010) are all framework based on the pairing based technique. The library function for the algorithm are written in C/C++, thus it finds wide spread application. However they are all designed for 8 bit platform which restricts there use to a definite kind of nodes only. Their usage on different heterogeneous WSN is different and thus their behaviour cannot be predicted. The data packet cannot maintain freshness in this technique.

Another such cryptography technique is the hybrid cryptographic technique which uses both asymmetric and symmetric encryption. Jailin et al., 2011 evolved a Dynamically Secured Authentication and Aggregation Scheme (DSAA) which uses both public and symmetric keys. The use of the same leads to better security and speed and reduces the use of memory. Different hybrid schemes are SCUR [80], MASA [81] and SecFleck [82]. SCUR tries to balance between the security requirement and the constraints of the WSN node. In MASA the private key is used for confidentiality, authentication and data integrity whereas event notification is authenticated by symmetric key. In MASA the maximum functionality is achieved by the asymmetric encryption. Thus this algorithm faces the same problem of high computational power requirement. SecFleck is based on a Trusted Platform Module chip which makes a node more trustworthy. However the algorithm is only built for Flake Sensors and thus the implementation is platform specific.

There are some specific security framework that is classified on the basis of the key management and the key distribution mechanism. Yong Wang et al,. [83] proposed a security framework UKEYING based on use of two different keys for MAC and for encryption. The calculation of the keys is dependent on the polynomial calculation and is very power intensive. This scheme generates high computational overhead which may not be desirable in most cases.

3. PROPOSAL OF FRAMEWORK

Most of the research community have utilised the WSN infrastructure as a random deployment pattern for military uses. Such random deployment poses several challenges to the end users. However there are some instances where the deployment pattern can be infrastructure based, like border fence monitoring, critical installation monitoring and mine field protection. This kind of planned deployment eases out some of the issues, like radio range and routing, for the network planner. However the importance of security mechanism cannot be overlooked.

We propose our security framework for such kind of planned deployment of WSN. We look to address the primary concern of security like data confidentiality, data integrity and message integration. We also try to thwart the replay attack and bring in a concept of timestamping. Node replication is addressed by the use of media access control address binding.

We plan to divide this security framework in to different modules like cryptographic algorithm, mode of operation, MAC and key management. The literature survey has been conducted as per the module. We intend to pick the most efficient algorithm or protocol for the respective module and base our framework accordingly. We try to strike a balance between the resource restrictions of WSN and provide a degree of reliable security.

For the cryptographic algorithm we select MISTY1 as the preferred method. MISTY1 is a symmetric block cipher and uses block size of 64 bits and key size of 128 bits. MISTY1 has proved its security against linear and differential cryptanalysis.

For the mode of operation for encryption we select the CTR mode of operation. CTR mode provides guaranteed provable-security. Though CTR may be only behind OFB in term of energy efficiency, yet the speed by which CTR gains synchronisation, we select CTR as the mode of operation.

In case of MAC, two methods stand to merit for selection, namely HMAC and CBCMAC. CBCMAC works on the underlying cipher which needs to be 128 bit block cipher like AES-128. This has a huge time penalty. However, this mode is preferred when the cryptographic method used is a 128 bit block cipher. This leads to automatic code minimization as the same cipher code is used for both encryption and MAC. But when the underlying block cipher is of 64 bit block size (like in the case of MISTY1), then it is logical to select a HMAC algorithm, which is not only faster in operations but also provide provable security. So we select HMAC as the message authentication code. We use HMAC with MD5 as the hashing function.

Key management in case of symmetric encryption is of prime importance as the same key is used for both encryption and decryption. The selection of key management mechanism partially depends on the deployment pattern. Since the deployment under consideration is an infrastructure based, we assume a trusted base station. The same concept finds implementation in SPINS, Perrig et al., 2001. However we propose to implement the concept of trusted base station as

Table 1: Comparison of Different Symmetric Key Framework

Framewo-rk	Encryption Used	Block Cipher mode	MAC	Key Management
SPINS	SNEP	Single block cipher	Broadcast authority (μTESLA)	Shared secret key with base station (Trusted Base Station)
LEAP	RC5	CBC	CBC-MAC	Initial key given by trusted base station. Multiple keys like group, cluster and pairwise shared key
TinySec	RC5/Skipjack/AES	CBC	CBC-MAC	Not mentioned. User has to device own method.
Proposed Framework	MISTY1	CTR	HMAC with MD5	Trusted base station with multiple authentication key

has been implemented by Chih-Chun Chang et al [84]. This method is an improvement over the key management concept used in SPINS. Chih-Chun Chang et al, proposes use of multiple authentication key in each node. The number of keys is dependent on the key length and the available space. It uses multiple base stations to ensure scalability and to negate the compromise or failure of a single base station.

We also propose to implement a time stamping on the message to ensure data freshness.

All the modern genre of nodes comes with a media access control address (MAC). We propose to use a MAC binding of the nodes at the base station. This will also enhance security and assist in finding malicious node.

In Table 1, we compare our proposed model with the available symmetric key security framework. Our proposed model is unique and provides a high degree of reliable security under the constraints of WSN.

4. METHODOLOGY FOR DEPLOYING

4.1 Assumptions.

Every framework being deployed is based on some primary assumptions. The assumptions are basically addressing those parameters of concerns which effect the functioning of the parameter at all others layers in the network stack. Here we are primarily concerned with the functioning of the framework at the application layer. Thus the security breaches possible due to the design flaws at the other layers are negated with the assumption that all other layers have been designed perfectly and there is no security design flaw at other layers. Sometimes such assumptions may not hold good but compartmentalisation of the problem statement provides better security and understanding. While designing this framework we have assumed certain parameters which do not affect the functioning at the application layer. They are as follows:

Initial deployment assumption: The primary assumption here is that the motes are deployed in an infrastructure based setup such that all the motes are within the extended communication range of the base station. The motes are randomly deployed with in the setup. This assumption can be extended to a hierarchical setup with several base stations being deployed within the extended communication range of a larger base station. This present setup being discussed is only considering one level of hierarchy.

Loose time synchronisation. It is assumed that the network maintains loose time synchronisation among all the nodes and the base station. This assumption is required for the effectiveness of the key management protocol that has been used.

No MAC addresses duplication. All the motes that are going to be used in this setup come with a hardwired MAC address. It is being assumed that no attack on this network is possible in term of MAC address duplication.

Existing protocol stack is functioning properly. We assume that all other layers in the protocol stack are functioning properly. The primary problem regarding the unreliability of wireless communication is assumed as fully reliable.

Trusted base station. Here while designing the framework we assume a trusted base station. We assume that the functioning of the base station is fully trusted and there are no communication breaches possible. Such as assumption is justified for a infrastructure based deployment as the deployment is planned and under supervision.

4.2 Phases of deployment

The methodology for deployment is done in a phased manner addressing one issue at a time and then designing the framework as a whole. The major part of the framework is based on the cryptographic algorithm. MISTY1 is coded in C language with an aim to optimise the code space as well as computational overhead. MISTY1 is a 64 bit block cipher thus limiting its use to only encryption. The same cannot be used for authentication mode of operations as it requires a cryptographic algorithm to be of minimum 128 bits. Cipher block chaining is carried as the second instance after the encryption module is found to be functioning properly.

The authentication mechanism is being taken care of by the Hash algorithm using MD5. As the encryption module and the authentication module are independent so they can be developed in parallel. The initial algorithm will be tested with the pre fed keys. Once the system is found to be working for the default set of keys, then the key management mechanism is incorporated. Since we are designing the system based on a trusted base station model so we incorporate the majority of task of key management at the base station.

The first phase of development incorporates the checking of the encryption and authentication mechanism in a single mode with the pre fed keys. The same is extended to communication between the base station and this node. Once the same is found to be operational then the key management module is introduced into the code and the pre fed keys are removed. The base station behaviour in key management is monitored.

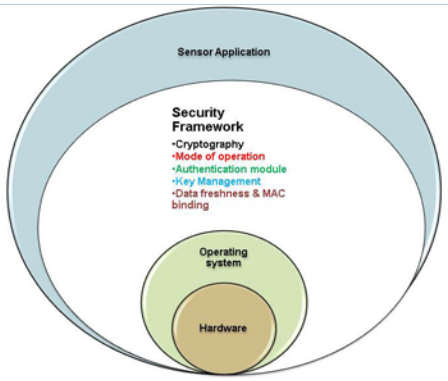

Fig 2. Development of framework for a single node

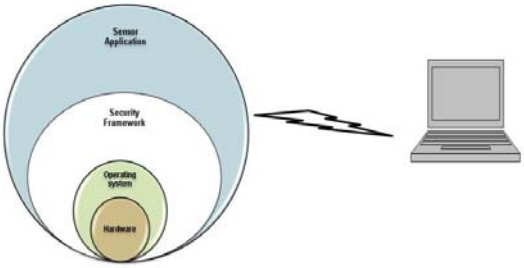

Fig 3. Base station communication with single mote

In the second phase of deployment the number of wireless station is increased to two. But they are made to communicate among themselves on a single hop communication. However in this case only the pre fed keys are tried out as the bulk of the key management functionalities are done by the base station and this scenario is devoid of the base station.

In the third phase we make the two wireless sensors communicate among themselves and also the base station. This scenario is tested with the key management algorithm in place. As we are only interested in deploying the security framework for a single hop situation, so we conclude our deployment with phase 3 of the trials.

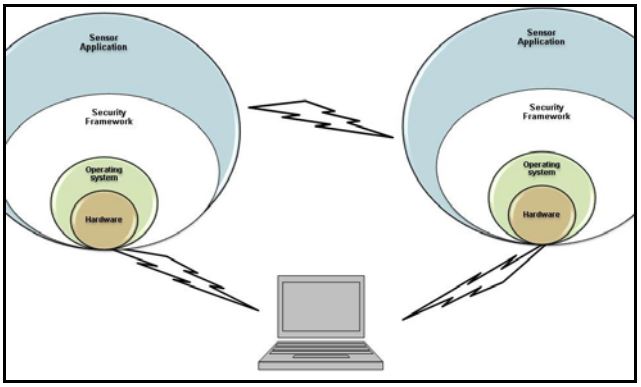

Fig 4. Base station communication with two motes

4.3 Software and hardware being used

The best way to demonstrate the functioning of this security framework under the set of assumptions is to make use of some real time simulators. A lot of simulators were studied for suitability. The major WSN operating system is TinyOS and Contiki OS. Both of these are open source and find wide spread use in many different motes. TinyOS platform motes need to be programmed in NesC language where as Contiki OS needs to be programmed in C/C++ language. There are several different simulators meant for each development platform OS. However TOSSIM for TinyOS and COOJA for Contiki OS stand out in term of functionality and ease of usage. Our implementation is being realised by the use of Contiki OS on COOJA simulators. The primary reason for making this choice is because of the ease of programming and the helpful visual guide provided in the simulator.

4.4 Measurement Matrices

The designed framework is being compared with the existing symmetric key security framework on the basis of code size, computational overhead and power consumption. Wherever a choice has to be made, security concerns have been given prime importance and has been vetoed against all other performance matrices. Since the framework is being deployed for mission critical applications, so security is of the utmost importance. Thus it has been given additional weightage over other performance matrices.

CONCLUSION

We conclude this paper with the future direction of work. Security has been the major guiding factor in designing this framework within the restriction laid down by WSN.

We have also tried to demonstrate the basic design consideration of a security framework for WSN.

Though in present context, security concern has been the driving force behind the proposed framework, it need not be the case always. One may try to design the framework on other efficiency parameters like code size minimization, computational efficiency or power consumption efficiency. The design consideration may change accordingly. This paper brings out the guidelines for selection of the algorithm or protocol. Any specific application requirement must be decided by the user as per the deployment plans.

REFERENCES

[1] J. Beutel, K. Römer, M. Ringwald, and M. Woehrle, 'Deployment Techniques for Sensor Networks', *Signals and Communication Technology*, pp. 219–248, Jan. 2009.

[2] M. Younis and K. Akkaya, 'Strategies and techniques for node placement in wireless sensor networks: A survey', *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, Jun. 2008.

[3] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, 1st ed. Boca Raton: CRC Press, 1997.

[4] R. Tripathi and S. Agrawal, 'Comparative Study of Symmetric and Asymmetric Cryptography Techniques', *International Journal of Advance Foundation and Research in Computer (IJAFRC)*, vol. 1, no. 6, pp. 68–76, Jun. 2014.

[5] R. Masram, V. Shahare, J. Abraham, and R. Moona, 'Analysis and Comparison of Symmetric Key Cryptographic Algorithms Based on Various File Features', *International Journal of Network Security & Its Applications*, vol. 6, no. 4, pp. 43–52, Jul. 2014.

[6] S.M. van den Broek, "Digital Signatures and the Public Key Infrastructure," PhD thesis, Erasmus University, Rotterdam, April 1999.

[7] W. Stallings, *Cryptography and Network Security: Principles and Practice*. India: Pearson Education India, 2011.

[8] B. Guido, L. Breveglieri, and M. Venturi, 'Power aware design of an elliptic curve coprocessor for 8 bit platforms', in *Fourth IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2006, pp. 337–342.

[9] D. J. Malan, M. Welsh, and M. D. Smith, 'A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography', in*First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2004, pp. 71–80.

[10] Piotrowski, Krzysztof, P. Langendoerfer, and S. Peter, 'How public key cryptography influences wireless sensor node lifetime', in *the fourth ACM workshop on Security of ad hoc and sensor networks*, 2006, pp. 169–176.

[11] M. Chowdhury, M. F. Kader, and A. Asaduzzaman, 'Security Issues in Wireless Sensor Networks: A Survey', *International Journal of Future Generation Communication and Networking*, vol. 6, no. 5, pp. 97–116, Oct. 2013.

[12] Federal Information Processing Standards Publication 197, 'Advanced Encryption Standard.' 2001.

[13] G. Jian, T. Peyrin, A. Poschmann, and R. Matt, 'The LED block cipher', in *Cryptographic Hardware and Embedded Systems–CHES 2011*, 2011, pp. 326–341.

[14] L. Knudsen and D. Wagner, 'On the structure of Skipjack', *Discrete Applied Mathematics*, vol. 111, no. 1–2, pp. 103–116, Jul. 2001.

[15] NIST Standard Publication, 'Skipjack and KEA algorithm specifications.' May-1998.

[16] Federal Information Processing Standards Publication, 'Data Encryption Standard.' 1977.

[17] B. Schneier, 'Description of a new variable-length key, 64-bit block cipher (Blowfish)', *Lecture Notes in Computer Science*, pp. 191–204, Jan. 1994.

[18] C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin, and H. Sibert, 'Sosemanuk, a Fast Software-Oriented Stream Cipher',*Lecture Notes in Computer Science*, pp. 98–118, Jan. 2008.

[19] Wu, Hongjun. "The stream cipher HC-128." *In New Stream Cipher Designs*, pp. 39-47. Springer Berlin Heidelberg, 2008.

[20] C. Cannière and C. De Cannière, 'Trivium: A Stream Cipher Construction Inspired by Block Cipher Design Principles', *Lecture Notes in Computer Science*, pp. 171–186, Jan. 2006.

[21] M. Boesgaard, M. Vesterager, T. Christensen, and E. Zenner, "The Stream Cipher Rabbit." [Online]. Available: http://www.ecrypt.eu.org/ stream/p3ciphers/rabbit/rabbit_p3.pdf.

[22] D. J. Bernstein, 'The Salsa20 Family of Stream Ciphers', *Lecture Notes in Computer Science*, pp. 84–97, Jan. 2008.

[23] S. Babbage and M. Dodd, 'The MICKEY Stream Ciphers', *Lecture Notes in Computer Science*, pp. 191–209, Jan. 2008.

[24] Gustavo Banegas, "Attacks in Stream Ciphers: A Survey", *International Association for Cryptologic Research*, 2014.

[25] Audia S. Abd Al-Rasedy, Ameer A.J Al-Swidi, "An advantages and Dis Advantages of Block and Stream Cipher", *Third Annual Scientific of the Faculty of basic Education Conference*, March 2002.

[26] M. Cazorla, K. Marquet, and M. Minier, " Survey and benchmark of lightweight block ciphers for wireless sensor networks," in *SECRYPT, P. Samarati, Ed. SciTePress*, 2013, pp. 543-548.

[27] Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata, "The 128-bit block cipher CLEAFIA", *Fast Software Encryption - FSE 2007*, Springer, pp. 181-195, 2007.

[28] Leander, Gregor, Christof Paar, Axel Poschmann, and Kai Schramm. "New lightweight DES variants." *In Fast Software Encryption*, pp. 196-210. Springer Berlin Heidelberg, 2007.

[29] Hong, Deukjo, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee et al. "HIGHT: A new block cipher suitable for low-resource device." *In Cryptographic Hardware and Embedded Systems-CHES 2006*, pp. 46-59. Springer Berlin Heidelberg, 2006.

[30] Lai, Xuejia, and James L. Massey. "A proposal for a new block encryption standard." *In Advances in Cryptology—EUROCRYPT'90*, pp. 389-404. Springer Berlin Heidelberg, 1991.

[31] De Canniere, Christophe, Orr Dunkelman, and Miroslav Knežević. "KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers*." In Cryptographic Hardware and Embedded Systems-CHES 2009*, pp. 272-288. Springer Berlin Heidelberg, 2009.

[32] Gong, Zheng, Svetla Nikova, and Yee Wei Law. "KLEIN: a new family of light weight blocks ciphers", Springer Berlin Heidelberg, 2012.

[33] Wu, Wenling, and Lei Zhang. "LBlock: a lightweight block cipher." In *Applied Cryptography and Network Security*, pp. 327-344. Springer Berlin Heidelberg, 2011.

[34] Guo, Jian, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. "The LED block cipher." In *Cryptographic Hardware and Embedded Systems–CHES 2011*, pp. 326-341. Springer Berlin Heidelberg, 2011.

[35] M. Izadi, B. Sadeghiyan, S. S. Sadeghian, and H. A. Khanooki, 'MIBS: A New Lightweight Block Cipher', in *Lecture Notes in Computer Science*, 2009, pp. 334–348.

[36] Daemen, Joan, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. "Nessie proposal: NOEKEON", In *First Open NESSIE Workshop*, pp. 213-230, 2000.

[37] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, 'Piccolo: An Ultra-Lightweight Blockcipher', in *Cryptographic Hardware and Embedded Systems – CHES 2011*, 2011, pp. 342–357.

[38] Bogdanov, Andrey, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe, "*PRESENT: An ultra-lightweight block cipher"*. Springer Berlin Heidelberg, 2007.

[39] Needham, R. M., and D. J. Wheeler. "Tea, a tiny encryption algorithm." In *Proceedings of the Second International Workshop on Fast Software Encryption (FSE 1994)*, vol. 1008, pp. 363-366. 1995.

[40] Suzaki, Tomoyasu, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. "TWINE: A Lightweight Block Cipher for Multiple Platforms." In *Selected Areas in Cryptography*, pp. 339-354. Springer Berlin Heidelberg, 2013.

[41] Standaert, François-Xavier, Gilles Piret, Neil Gershenfeld, and Jean-Jacques Quisquater. "SEA: A scalable encryption algorithm for small embedded applications." In *Smart Card Research and Advanced Applications*, pp. 222-236. Springer Berlin Heidelberg, 2006.

[42] Jeong, Kitae, HyungChul Kang, Changhoon Lee, Jaechul Sung, and Seokhie Hong. "Biclique Cryptanalysis of Lightweight Block Ciphers PRESENT, Piccolo and LED." *IACR Cryptology ePrint Archive, Vol 621*, 2012.

[43] Chen, Jiazhe, Meiqin Wang, and Bart Preneel. "Impossible differential cryptanalysis of the lightweight block ciphers TEA, XTEA and HIGHT." In *Progress in Cryptology-AFRICACRYPT 2012*, pp. 117-137. Springer Berlin Heidelberg, 2012.

[44] Sasaki, Yu, Lei Wang, Yasuhide Sakai, Kazuo Sakiyama, and Kazuo Ohta. "Three-subset meet-in-the-middle attack on reduced XTEA." In *Progress in Cryptology-AFRICACRYPT 2012*, pp. 138-154. Springer Berlin Heidelberg, 2012.

[45] Çoban, Mustafa, Ferhat Karakoç, and Özkan Boztaş. "Biclique cryptanalysis of TWINE." In *Cryptology and Network Security*, pp. 43-55. Springer Berlin Heidelberg, 2012.

[46] Bogdanov, Andrey, Dmitry Khovratovich, and Christian Rechberger. "Biclique cryptanalysis of the full AES." In *Advances in Cryptology–ASIACRYPT 2011*, pp. 344-371. Springer Berlin Heidelberg, 2011.

[47] Derbez, Patrick, Pierre-Alain Fouque, and Jérémy Jean. "Improved key recovery attacks on reduced-round AES in the single-key setting." In *Advances in Cryptology–EUROCRYPT 2013*, pp. 371-387. Springer Berlin Heidelberg, 2013.

[48] Y. Wei, J. Doumen, and P. Hartel, 'Survey and benchmark of block ciphers for wireless sensor networks', in *ACM Transactions on Sensor Networks (TOSN)*, 2006, vol. 2, no. 1, p. 65.

[49] R. L. Rivest, 'The RC5 encryption algorithm', in *Lecture Notes in Computer Science*, 1995, pp. 86–96.

[50] R. Rivest, M. J. B. Robshaw, R. Sidney, and Y. L. Yin, "The RC6&trade, block cipher", *Proc. 1st Advanced Encryption Standard (AES) Conf*, 1998.

[51] "*KASUMI Specification*", ETSI/SAGE Specification Version: 1.0, 1999.

[52] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, "Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms", in *Proc. Selected Areas in Cryptography (SAC'00)*, pp 39-56. Springer-Verlag, 2001.

[53] Matsui, Mitsuru. "New block encryption algorithm MISTY." In *Fast Software Encryption*, pp. 54-68. Springer Berlin Heidelberg, 1997.

[54] "*Recommendation for Block Cipher Modes of Operation*", NIST Standards 800-38A, 2001.

[55] Rogaway, Phillip. "Evaluation of some block cipher modes of operation." *Cryptography Research and Evaluation Committees (CRYPTREC) for the Government of Japan,* 2011.

[56] S. Goldwasser and S. Micali, 'Probabilistic encryption', *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, Apr. 1984.

[57] C. E. Shannon, 'Communication Theory of Secrecy Systems*', *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[58] J. Lee, K. Kapitanova, and S. H. Son, 'The price of security in wireless sensor networks', *Computer Networks*, vol. 54, no. 17, pp. 2967–2978, Jan. 2010.

[59] K. Brincat and C. J. Mitchell, 'New CBC-MAC Forgery Attacks', *Lecture Notes in Computer Science*, pp. 3–14, Jan. 2001.

[60] A. Joux, G. Poupard, and J. Stern, 'New Attacks against Standardized MACs', *Lecture Notes in Computer Science*, pp. 170–181, 2003..

[61] Mitchell, Chris J. "On the security of XCBC, TMAC and OMAC." *Comments to NIST, August* 19, 2003.

[62] M. Bellare, R. Canetti, and H. Krawczyk, 'Keying Hash Functions for Message Authentication', *Advances in Cryptology — CRYPTO '96*, pp. 1–15, 1996.

[63] Y. Dodis, R. Gennaro, J. Håstad, H. Krawczyk, and T. Rabin, 'Randomness Extraction and Key Derivation Using the CBC, Cascade and HMAC Modes', *Advances in Cryptology – CRYPTO 2004*, pp. 494–510, Jan. 2004.

[64] B. Preneel and P. C. van Oorschot, 'MDx-MAC and Building Fast MACs from Hash Functions', *Advances in Cryptology — CRYPT0' 95*, pp. 1–14, 1995.

[65] Contini, Scott, and Yiqun Lisa Yin. "Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions." In *Advances in Cryptology–ASIACRYPT 2006*, pp. 37-53. Springer Berlin Heidelberg, 2006.

[66] D. A. McGrew and J. Viega, 'The Security and Performance of the Galois/Counter Mode (GCM) of Operation', *Progress in Cryptology - INDOCRYPT 2004*, pp. 343–355, Jan. 2005.

[67] Carter, J. Lawrence, and M. N. Wegman, 'Universal classes of hash functions (Extended Abstract)', p. 106, Apr. 1977.

[68] Ferguson, Niels. "Authentication weaknesses in GCM." *Comments submitted to NIST Modes of Operation Process*, 2005.

[69] Fouque, Pierre-Alain, Gwenaëlle Martinet, Frédéric Valette, and Sébastien Zimmer. "On the Security of the CCM Encryption Mode and of a Slight Variant." In *Applied Cryptography and Network Security*, pp. 411-428. Springer Berlin Heidelberg, 2008.

[70] Raazi, Syed Muhammad Khaliq-ur-Rahman, Zeeshan Pervez, and Sungyoung Lee. "Key management schemes of wireless sensor networks: A survey." *Department of Computer Engineering, Kyung Hee University, Global Campus, Korea*, 2011.

[71] Bala, Suman, Gaurav Sharma, and Anil K. Verma. "Classification of symmetric key management schemes for wireless sensor networks." *International Journal of Security and Its Applications* 7, vol. 2, pp 117-138, 2013.

[72] Chang, Chih-Chun, Shadi Arafa, and Sead Muftic. "Key establishment protocol for wireless sensor networks." In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, pp. 1-6. IEEE, 2007.

[73] L. J. C, L. V. C. M, W. K. H, C. Jiannong, and C. H. C. B, 'Key management issues in wireless sensor networks: current proposals and future developments', *Wireless Communications, IEEE*, vol. 14, no. 5, pp. 76 – 84, Oct. 2007.

[74] G. Sharma, S. Bala, and A. K. Verma, 'Security Frameworks for Wireless Sensor Networks-Review', *Procedia Technology*, vol. 6, pp. 978–987, 2012.

[75] S. Zhu, S. Setia, and S. Jajodia, 'LEAP+: Efficient security mechanisms for large-scale distributed sensor networks', in *ACM Transactions on Sensor Networks (TOSN)*, 2006, vol. 2, no. 4, p. 500.

[76] Karlof, Chris, Naveen Sastry, and David Wagner. "TinySec: a link layer security architecture for wireless sensor networks." In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 162-175. ACM, 2004.

[77] Watro, Ronald, Derrick Kong, Sue-fen Cuti, Charles Gardiner, Charles Lynn, and Peter Kruus. "TinyPK: securing sensor networks with public key technology." In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pp. 59-64. ACM, 2004.

[78] Liu, An, and Peng Ning. "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks." In *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*, pp. 245-256. IEEE, 2008.

[79] Y. Wang , G. Attebury and B. Ramamurthy "A survey of security issues in wireless sensor networks", *IEEE Commun. Surveys Tutorials*, vol. 8, pp.2 -23 2006..

[80] Ruhma Tahir, Muhammad Y. Javed, Attiq Ahmad and Raja Iqbal, "SCUR: Secure Communications in Wireless Sensor Networks using Rabbit", Proceedings of the World Congress on Engineering 2008, Vol I, WCE 2008, July 2 - 4, 2008, London, U.K.

[81] Alzaid, Hani, and Manal Alfaraj. "MASA: End-to-End Data Security in Sensor Networks Using a Mix of Asymmetric and Symmetric Approaches." In *New Technologies, Mobility and Security, 2008. NTMS'08*, pp. 1-5. IEEE, 2008.

[82] Hu, Wen, Peter Corke, Wen Chan Shih, and Leslie Overs. "secfleck: A public key technology platform for wireless sensor networks." In *Wireless Sensor Networks*, pp. 296-311. Springer Berlin Heidelberg, 2009.

[83] Wang, Yong, Byrav Ramamurthy, Yuyan Xue, and Xukai Zou. "A security framework for wireless sensor networks utilizing a unique session key." In *Broadband Communications, Networks and Systems, 2008. BROADNETS 2008. 5th International Conference on*, pp. 487-494. IEEE, 2008.

[84] Chang, Chih-Chun, Shadi Arafa, and Sead Muftic. "Key establishment protocol for wireless sensor networks." *In Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, pp. 1-6. IEEE, 2007.

Sudipto Roy received his B Tech degree from Military College of Telecommunication Engineering, Mhow, India. He is currently undergoing his MTech degree in Cyber Security from Defense Institute of Advanced Technology (DIAT), Pune. He has an active service of 10 years in Indian Army. His areas of interest are Adhoc Networks, Cryptography, WSN and wireless communications.

Manisha J. Nene received the Ph.D. degree in Computer Science and Engineering from the Defense Institute of Advanced Technology (DIAT), Pune; a Defense Research and Development Organization (DRDO) Establishment under the Ministry of Defense, India. She is currently a Faculty Member with the Department of Computer Science and Engineering, DIAT, Pune. Her areas of interest are cyber physical systems, wireless sensor networks, analysis of algorithms, high performance computing, modeling, and simulations.