International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 7 Issue 1 –MARCH 2014.

EXTRACTING SECRET KEY FROM RECEIVED SIGNAL STRENGTH (RSS) MEASUREMENTS IN WIRELESS NETWORKS

R. Surya#, S.V. Manikanthan*

*PG Scholar #, Asst. Prof.**

Department of Electronics & Communication,

Dr.Pauls Engineering College, India.

Abstract- Evaluate the effectiveness of secret key extraction, for private communication between two wireless devices, from the received signal strength (RSS) variations on the wireless channel between the two devices. In a variety of environments, RSS may vary due to lack of variations in the wireless channel; the extracted bits have very low entropy making these bits unsuitable for a secret key, In significant movement environment, high entropy bits are obtained fairly quickly. Develop an environment adaptive secret key generation scheme that uses an adaptive lossy quantizer in conjunction with Cascade-based information reconciliation and privacy amplification. Our measurements show that our scheme, in comparison to the existing ones that we generate high entropy bits at a high bit rate. The secret key bit streams generated by Adaptive secret bit generation (ASBG) scheme. Finally, we evaluate secret key extraction in a multiple input multiple output (MIMO)-like sensor network testbed that we create using multiple TelosB sensor nodes. Our MIMO-like sensor environment produces prohibitively high bit mismatch, which we address using an iterative distillation stage that we add to the key extraction process. Ultimately, we show that the secret key generation rate is increased when multiple sensors are involved in the key extraction process.

Keywords—Adaptive secret bit generation, Received signal strength, TelosB sensor node, Cryptography

I. INTRODUCTION

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting.

A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

To evaluate the effectiveness of secret key extraction, for private communication between two wireless devices, from the received signal strength (RSS) variations. It provides the security using Secret Key Extraction mechanism for Wireless Devices during the data transmission. Secret key establishment is a fundamental requirement for private communication between two entities. Currently, the most common method for establishing a secret key is by using public key cryptography. However, public key cryptography consumes significant amount of computing resources and power which might not be available in certain scenarios (e.g., sensor networks).

More importantly, concerns about the security of public keys in the future have spawned research on methods that do not use public keys. Quantum cryptography is a good example of an innovation that does not use public keys. It uses the laws of Quantum theory, specifically Heisenberg's uncertainty principle, for sharing a secret between two end points. Although quantum cryptography applications have started to appear recently, they are still very rare and expensive.

Most of the previous research work on RSS-based secret key extraction, including that is based on either simulations or theoretical analysis. Building the strengths of the existing schemes, developing an environment Adaptive secret key generation scheme that uses an Adaptive Lossy quantizer in conjunction with Cascade-based information reconciliation and privacy amplification.

II. EXSISTING MODELS

The broadcast nature of a wireless link provides a natural eavesdropping and intervention capability to an adversary. Thus, securing a wireless link is essential to the security of a wireless network, and key generation algorithms are necessary for securing wireless links. However, Traditional

key agreement algorithms can be very costly in many settings, e.g. in wireless ad-hoc networks, since they consume scarce resources such as bandwidth and battery power.

In this paper [04], establishing a novel approach that couples the physical layer characteristics of wireless networks with key generation algorithms. It is based on the wireless communication phenomenon known as the principle of reciprocity which states that in the absence of interference both transmitter and receiver experience the same signal envelope. The key-observation here is that the signal envelope information can provide to the two transceivers two correlated random sources that provide sufficient amounts of entropy which can be used to extract a cryptographic key.

Confidential communication over wireless channels are established, the fundamental security limits of quasi-static fading channels from the point of view of outage secrecy capacity with perfect and imperfect channel state information. Then develop a practical secret key agreement protocol for Gaussian and quasi-static fading wiretap channels, that performs close to the fundamental secrecy capacity limits[8]. The communication is from the transmitter-to-receiver only and requires no feedback or error-free side channels. The key generation/distribution problem in wiretap channels falls under the general problem of key generation from correlated source outputs. Development of new security and communication metrics such as average throughput for average secret and non-secret bits, transmitted per channel use on the wiretap channel.

The protocol uses a four-step procedure to secure communications. Establish common randomness via an opportunistic transmission, perform message reconciliation, establish a common key via privacy amplification, and use of the key. Introducing a new reconciliation procedure that uses multilevel coding and optimized low density parity check codes which in some cases comes close to achieving the secrecy capacity limits. Finally, develop a new metrics for assessing average secure key generation rates and show that our protocol is effective in secure key renewal.Random number sequences are of crucial importance in every aspect of modern digital cryptography, having a significant impact on the strength of cryptographic primitives in securing secret information by rendering it unknown, ungues sable, unpredictable and irreproducible for an adversary [1]. In this context, the present paper aims to accentuate the crucial importance of random numbers in cryptography and to suggest a set of efficient and practical methods for the generation and testing of random number sequences intended for cryptographic applications.

III. SYSTEM METHODOLOGY

In this section, we first describe the three components of our wireless RSS-based secret key extraction. Next, we briefly describe two classes of existing quantization approaches. Last, we develop a new approach by combining the advantages of the existing approaches.

A. Components of Secret Key Extraction

To establish a shared secret key, Alice and Bob measure the variations of the wireless channel between them across time by sending probes to each other and measuring the RSS values of the probes. Ideally, both Alice and Bob should measure the RSS values at the same time. However, typical commercial wireless transceivers are half duplex, i.e., they cannot both transmit and receive the signals simultaneously. Thus, Alice and Bob must measure the radio channel in one direction at a time. However, as long as the time between two directional channel measurements is much smaller than the inverse of the rate of change of the channel, they will have similar RSS estimates. Most of the existing literature on key extraction from RSS measurements either use some or all of the following three steps.

1. *Quantization*

As multiple packets are exchanged between Alice and Bob, each of them builds a time series of measured RSS. Then, each node quantizes its time series to generate an initial secret bit sequence. The quantization is done based on specified thresholds. Fig. 1 shows a sample RSS quantizer with two thresholds. Different quantizers have been proposed in the existing literature [5]. The difference in these quantizers mainly results from their different choices of thresholds and the different number of thresholds that they use.

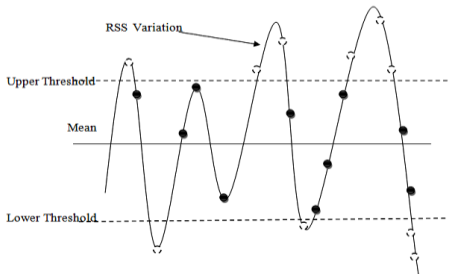

Fig. 1: A sample RSS quantizer with two thresholds

2. *Information Reconciliation*

Once both Alice and Bob extract the bit stream from the RSS measurements they collect using quantizers, to agree upon the same key, they must correct the bits where the two bit streams differ. Differences in their bit streams primarily arise due to the following—presence of noise and interference, hardware limitations, manufacturing variations, vendor-specific differences including differences in implementing automatic gain control, and the lack of sampling at the same time at Alice and Bob, primarily due to the half-duplex mode of communication in commercial transceivers. Cascade [2] is an iterative, interactive information reconciliation protocol. In this protocol, Alice permutes the bitstream randomly, divides it into small blocks, and sends permutation and parity information of each block to Bob. Bob permutes his bitstream in the same way, divides it into small blocks, computes parities and

checks for parity mismatches. For each mismatch, Bob performs a binary search on the block to find if a few bits can be changed to make the block match the parity. These steps are iterated a number of times to ensure a high probability of success.

3. *Privacy Amplification*

When the probe packets are exchanged at a rate greater than the inverse of the coherence interval of the channel, there may be short-term correlation between subsequent quantized bits. Moreover, the information reconciliation stage reveals a certain fraction of information to correct the mismatching bits of Alice and Bob; the leaked portion needs to be removed so that an adversary cannot use this information to guess portions of the extracted key. Privacy amplification solves the above two problems by reducing the size of output bit stream. This is achieved by letting both Alice and Bob use universal hash functions, chosen at random from a publicly known set of such functions, to obtain fixed size smaller length output from longer input streams. Essentially, privacy amplification generates a shorter secret bit stream with a higher entropy rate from a longer secret bit stream with a lower entropy rate. Most of the popular methods used for privacy amplification are based on the leftover hash lemma, a well-known technique to extract randomness from imperfect random sources.

B. Adaptive Secret Bit Generation (ASBG)

The RSS measurements do not generate secret bits at a high rate and/or with high entropy. A method, that we call ASBG, that builds on the strengths of the existing approaches. In our method, we use a modified version of Mathur's quantizer in conjunction with two well-known information reconciliation and privacy amplification techniques.

We first describe our quantizer and then identify the differences with Mathur's scheme. Our modified quantizer is described as follows: 1) Alice and Bob consider a block of consecutive measurements of size block_size which is a configurable parameter. For each block, they calculate two adaptive thresholds q+ and q- independently such that q+= mean +α * std_deviation and q-= mean –α * std_deviation, where $\alpha \geq 0$. 2) Alice and Bob parse their RSS measurements and drop RSS estimates that lie between q+ and q- and maintain a list of indices to track the RSS estimates that are dropped. They exchange their list of dropped RSS estimates and only keep the ones that they both decide not to drop. 3) Alice and Bob generate their bit streams by extracting a 1 or a 0 for each RSS estimate if the estimate lies above q+ or below q-, respectively. Our modified quantizer divides the RSS measurements into smaller blocks of size block_size and calculates the thresholds for each block separately. The adaptive thresholds allow our quantizer to adapt to slow shifts of RSS. Mathur et al. subtract a running windowed average of RSS measurements before computing thresholds q + and q - to make their scheme adaptive to the slow variations of RSS. We also perform experiments to find the optimal block size. The results of these experiments are shown in Section 6. Unlike the Mathur quantizer that preserves only a single bit from m consecutive 1s or 0s and drops the other repeating m _ 1 bits, our modified quantizer extracts a bit out of each measurement that falls above the upper threshold or below the lower threshold but depends on the privacy amplification step to remove the effect of correlated bits.

C. Random Number Generation

A random number generator (RNG) is a computational or physical device designed to generate a sequence of numbers or symbols that lack any pattern, i.e. appear random. The many applications of randomness have led to the development of several different methods for generating random data. Many of these have existed since ancident times, including dice, coin flipping, the shuffling of playing cards, the use of yarrow stalks (by divination) in the I Ching, and many other techniques. Because of the mechanical nature of these techniques, generating large numbers of sufficiently random numbers (important in statistics) required a lot of work and/or time. Thus, results would sometimes be collected and distributed as random number tables. Nowadays, after the advent of computational random number generators, a growing number of government-run lotteries, and lottery games, are using RNGs instead of more traditional drawing methods. RNGs are also used today to determine the odds of modern slot machines.

Here the source sends a data to the destination; data is forwarded to the intermediate nodes one by one, based on Received Signal Strength (RSS). Secret Key is generated which is passed to both the Source and the Destination. Destination node concatenated the intermediate node keys and generates the four digit random key. Destination node sends the random key to source node via intermediate node. The source node already has intermediate node keys. So, it verifies the random key and identifies the destination node is correct or not. Source concatenated the intermediate node keys to destination via intermediate nodes. Destination node verifies the random key and identify the source node is correct or not.

D. Hash Code Generation

Hash functions are primarily used to generate fixed-length output data that acts as a shortened reference to the original data. This is useful when the output data is too cumbersome to use in its entirety.

One practical use is a data structure called a hash table where the data is stored associatively. Searching for a person's name in a list is slow, but the hashed value can be used to store a reference to the original data and retrieve constant time (barring collisions). Another use is in cryptography, the science of encoding and safeguarding data. It is easy to generate hash values from input data and easy to verify that the data matches the hash, but hard to 'fake' a hash value to hide malicious data. This is the principle behind the Pretty Good Privacy algorithm for data validation.

Hash functions are also used to accelerate table lookup or data comparison tasks such as finding items in a database, detecting duplicated or similar records in a large file, finding similar stretches in DNA sequences, and so on. A hash function should be referentially transparent (stable), i.e., if called twice on input that is "equal" (for example, strings that consist of the same sequence of characters), it should give the same result. There is a construct in many programming languages that allows the user to override equality and hash functions for an object: if two objects are equal, their hash codes must be the same. This is crucial to finding an element in a hash table quickly, because two of the same element would both hash to the same slot. Hash functions are destructive, akin to lossy compression, as the original data is lost when hashed. Unlike compression algorithms, where something resembling the original data can be decompressed from compressed data, the goal of a hash value is to uniquely identify a reference to the object so that it can be retrieved in its entirety.

IV. DESCRIPTION OF ALGORITHM

A. *RSA Encryption*

RSA encryption is a public-key encryption technology developed by RSA Data Security. The RSA algorithm is based on the difficulty in factoring very large numbers. Based on this principle, the RSA encryption algorithm uses prime factorization as the trap door for encryption. Deducing an RSA key, therefore, takes a huge amount of time and processing power. RSA is the standard encryption method for important data, especially data that's transmitted over the Internet. RSA stands for the creators of the technique, Rivest, Shamir and Adelman.

B. *RSA Security Process*

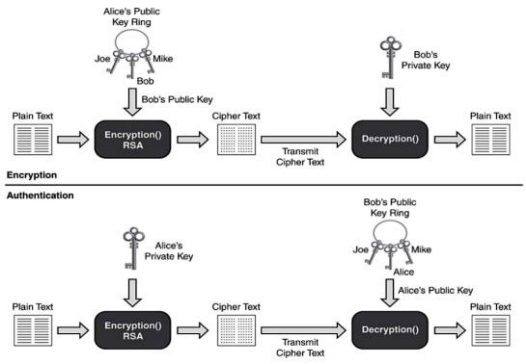

Fig. 2: RSA Security Process

Cryptographic methods cannot be proven secure. Instead, the only test is to see if someone can figure out how to decipher a message without having direct knowledge of the decryption key. The RSA method's security rests on the fact that it is extremely difficult to factor very large numbers. If 100 digit numbers are used for *p* and *q*, the resulting *n* will be approximately 200 digits. The fastest known factoring algorithm would take far too long for an attacker to ever break the code. Other methods for determining *d* without factoring *n* are equally as difficult. Any cryptographic technique which can resist a concerted attack is regarded as secure. At this point in time, the RSA algorithm is considered secure.

C. *Key Encription Algorithm*

First we need to define a directory that contains all the documents to be proceeding. There documents with extension ".txt", contains the stop words which has English as default language. Language can be changed according to the requirement. If one is using Free Indexing, than no need for vocabulary but in case of Controlled Indexing one has to define the vocabulary. KEA matches the document's phrases against the file. Length of Key phrases are pre-defined, Key phrases do not start or end with stop word defined in the vocabulary. In case of Controlled Indexing, it collects only those words which are matched with the thesaurus. If the thesaurus defines relations between non-allowed terms (non-descriptors) and allowed terms (descriptors).

Before being able to extract key phrases from new documents, KEA first needs to create a model, which tells the extraction strategy. This means, for each document in the input directory there must be a file with the extension ".key" and the same name as the corresponding document. This file should contain manually assigned key phrases, one per line. Given the list of the candidate phrases, KEA marks those that were manually assigned as positive example and all the rest as negative examples. By analyzing the feature values for positive and negative candidate phrases, a model is computed, which reflects the distribution of feature values for each phrase.

The final step is extraction of key phrase. Once the model has been build, according to the extraction policy defined in the model, key phrase are started to be extracted from the defined documents. According to that measure final list of Key phrases been retrieved. The number of how much key phrases be extracted can be controlled at the development time. Thus, KEA algorithm is mostly used for the Auto-Extraction of Keywords for metadata or any other purpose. The secret key generated by our scheme also passed the randomness tests We were able to further enhance the rate of secret bit generation of our scheme by extracting multiple bits from each RSS measurement. Secret key extraction in sensor network test bed and showed that secret key generation rate can be improved by involving multiple sensors in the key extraction process.

D. *Received Signal Strength*

Received signal strength (RSS) is a popular statistic of the radio channel and can be used as the source of secret information shared between a transmitter and receiver. We use RSS as a channel statistic, primarily because of the fact that most of the current of-the-shelf wireless cards, without any modification, can measure it on a per frame basis. Thev ariation over time of the RSS, which is caused by motion and multipath fading, can be quantized and used for generating secret keys. The mean RSS value, a somewhat predictable function of distance, must be filtered out of the

measured RSS signal to ensure that an attacker cannot use the knowledge of the distance between key establishing entities to guess some portions of the key.

We have two lateration techniques (ToA and TDoA) that use elapsed time to measure distance. Lateration can also be performed by using received signal strength (RSS) in place of time. With this approach, RSS is measured by either the mobile device or the receiving sensor. Knowledge of the transmitter output power, cable losses, and antenna gains as well as the appropriate path loss model allows you to solve for the distance between the two stations.

The following is an example of a common path loss model used for indoor propagation:

$$PL = PL_1 + 10\log(d^n) + s$$

In this model:

- *PL* represents the total *path loss* experienced between the receiver and sender in dB. This will typically be a value greater than or equal to zero.
- PL_{1Meter} represents the *reference path loss* in dB for the desired frequency when the receiver-to-transmitter distance is 1 meter. This must be specified as a value greater than or equal to zero.
- *d* represents the *distance* between the transmitter and receiver in meters.
- *n* represents the *path loss exponent* for the environment.
- *s* represents the standard deviation associated with the degree of *shadow fading* present in the environment, in dB. This must be specified as a value greater than or equal to zero.

Path loss (*PL*) is the difference between the level of the transmitted signal, measured at face of the transmitting antenna, and the level at of the received signal, measured at the face of the receiving antenna. Path loss does not take antenna gains or cable losses into consideration. Path loss represents the level of signal attenuation present in the environment due to the effects of free space propagation, reflection, diffraction, and scattering.

The generally accepted method to calculate receiver signal strength given known quantities for transmit power, path loss, antenna gain, and cable losses is as follows:

$$RX_{PWR} = TX_{PWR} - Loss_{TX} + Gain_{TX} - PL_1 + Gain_{RX} - LOSS_{RX}$$

We can directly substitute our equation for path loss into the equation above. This enables us to solve for distance *d* as follows:

$$d = 10^{\frac{TX_{PWR} - RX_{PWR} - Loss_{TX} + Gain_{TX} - PL_1 + S + Gain_{RX} - LOSS_{RX}}{10^n}}$$

where the meaning of the terms in the equation above are:

- Rx_{PWR} represents the detected receive signal strength in dB.
- Tx_{PWR} represents the transmitter output power in dB.
- $Loss_{TX}$ represents the sum of all transmit-side cable and connector losses in dB.
- $Gain_{TX}$ represents the transmit-side antenna gain in dBi.
- $Loss_{RX}$ represents the sum of all receive-side cable and connector losses in dB.
- $Gain_{RX}$ represents the receive-side antenna gain in dBi.

E. RSA Key Generation Algorithm

The RSA algorithm is used for both public key encryption and digital signatures. It is the most widely used public key encryption algorithm. The basis of the security of the RSA algorithm is that it is mathematically infeasible to factor sufficiently large integers.

The RSA algorithm is believed to be secure if its keys have a length of at least 1024-bits.

1. Choose two very large random prime integers:
p and q
2. Compute n and φ(n):
n = pq and φ(n) = (p-1)(q-1)
3. Choose an integer e, 1 < e < φ(n) such that:
gcd(e, φ(n)) = 1(where gcd means greatest common denominator)
4. Compute d, 1 < d < φ(n) such that:
ed ≡ 1 (mod φ(n))

- the public key is (n, e) and the private key is (n, d)
- the values of p, q and φ(n) are private
- e is the public or encryption exponent
- d is the private or decryption exponent

Encryption-The cipher text C is found by the equation 'C = M^e mod n' where M is the original message.
Decryption-The message M can be found form the cipher text C by the equation 'M = C^d mod n'.

V. RESULT AND DISCUSSION

A. Resultant output from NS2

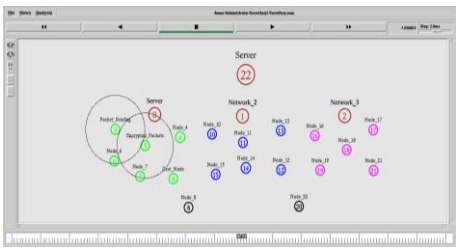

Fig. 3: Transmission of Encrypted packets

The fig.3 shows the Transmission of encrypted packets from one Network node to another Network node.

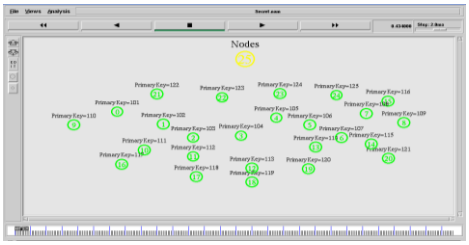

Fig. 4: Primary key Generation.

The Fig.4 explains the Primary key generation for encrypted packets.

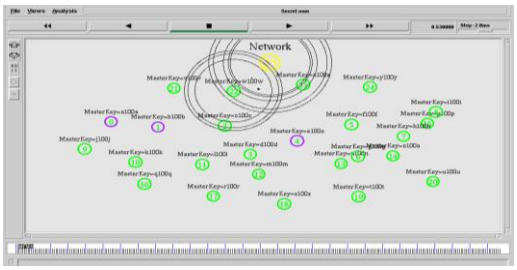

Fig. 5: Master key Generation.

The Fig.5 explains the Master key generation after verifying the primary key.

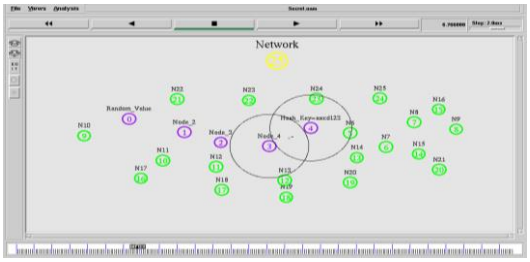

Fig. 6: Hash Code Generation

The Fig.6 explains the Hash code Generation after generating the Master key.

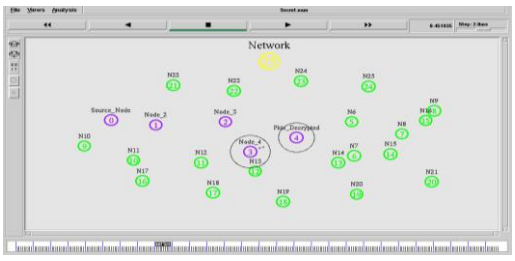

Fig. 7: Packets reaches the Destination.

The Fig.7 explains the encrypted packets generating the primary key, master key, hash code and then reaches the destination.

B. Conclusion and Future Enhancement

The rate of secret bit generation of our scheme by extracting multiple bits from each RSS measurement. We also evaluated secret key extraction in a MIMO-like sensor network testbed and showed that secret key generation rate can be improved by involving multiple sensors in the key extraction process. The predictable channel attack are primarily for key extraction using RSS measurements and these may not directly applyto key extraction using channel impulse response measurements.

Extract the secret key from RSS measurements. Implementing a strong Verification Scheme in the Destination End. Destination's User Name, Password, IP Address, Primary Key, Parsed Random Key, Hash Value of Secret Key, Decryption Key to open the Data, as well as Secondary Key for changing the Primary key are all verified for the Secured Communication of Data between Source and any Destination.

REFERENCES

[1] "NIST, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," http:// csrc.nist.gov/publications/nistpubs/800-22/sp-800-22-051501. pdf, 2001.

[2] "ipwraw,"http://homepages.tu darmstadt.de/~p_larbig/wlan, 2012.

[3] "Radiotap," http://www.radiotap.org, 2012.

[4] "ConvertingSignal Strength Percentage to dBm Values," http://www.wildpackets.com/elements/whitepapers/Converting_ Signal_Strength.pdf, 2012.

[5] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless Secret Key Generation Exploiting Reactance-Domain Scalar Response of Multipath Fading Channels,"IEEE Trans. Antennas and Propagation, vol. 53, no. 11, pp. 3776-3784, Nov. 2005.

[6] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust Key Generation from Signal Envelopes in Wireless Networks," Proc. 14th ACM Conf. Computer and Comm. Security (CCS), 2007.

[7] C.H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental Quantum Cryptography," J. Cryptology, vol. 5, no. 1, pp. 3-28, 1992.

[8] M. Bloch, J. Barros, M.R.D. Rodrigues, and S.W. McLaughlin,"Wireless Information-Theoretic Security," IEEE Trans. InformationTheory, vol. 54, no. 6, pp. 2515-2534, June 2008.

[9] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N.B.Mandayam, "Information-Theoretically Secret Key Generation forFading Wireless Channels," IEEE Trans. Information Forensics andSecurity, vol. 5, no. 2, pp. 240-254, June 2010.

Ms.R.SURYA has been received the BE degree in Computer Science Engineering in 2012.Currently she is pursuing her M.E degree in Computer and Communication Engineering .Her area of interest are Network Security .

S.V.Manikanthan, was born in Chennai, Indiain . He received his Diploma of Engineering in Electronics and

Communication Engineering, from DOTE, Chennai and Bachelor of Engineering degree in Electronics and Communication from University of Madras, Chennai, India. Master of Engineering in Communication Systems from University of Vels, Chennai, Currently he is working as Assit.Professor in Dr. Pauls Engineering College, Villupuram and affiliated to Anna University Chennai, Tamil Nadu, India. His area of interest includes Wireless Sensor and Computer Networks.